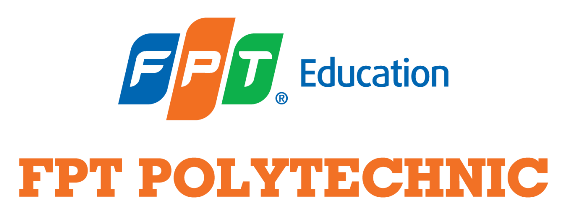
**TRƯỜNG CAO ĐẲNG THỰC HÀNH**

**FPT POLYTECHNIC - TP. HCM**

🞟🞟🕮🞟🞟

****

**DỰ ÁN 1**

**NGÀNH CNTT (ỨNG DỤNG PHẦN MỀM)**

**PHẦN MỀM QUẢN LÝ SIÊU THỊ MINI**

***“SuperSALES”***

***Phần 01***

SVTH: 1. Nguyễn Trần Duy – PS09687 (Trưởng nhóm)

2. Phan Văn Phú – PS08761

3. Nguyễn Quốc Khánh – PS08765

4. Nguyễn Nhật Thông – PS08608

LỚP: UD14306

GVHD: ThS. Lê Anh Tú

**TPHCM –2019**

**LỜI NÓI ĐẦU**

Với mọi ngành nghề công việc, kỹ năng cứng và kỹ năng mềm là hai yêu cầu luôn đi đôi với nhau không tách rời. Đặc biệt đối với lập trình viên, kỹ thuật phân tích thiết kế hệ thống và kỹ năng năng lập kế hoạch, làm việc nhóm là rất quan trọng. Vì thế, chúng em tiếp cận môn *“Dự án 1”* với hai mục tiêu cơ bản: 1 – ôn tập và đào sâu những kiến thức đã học về lập trình Java, 2 – vận dụng kỹ năng làm việc nhóm vào công việc thực tế.

Chúng em lựa chọn xây dựng phầm mềm quản lý siêu thị mini, bao gồm các yêu cầu nghiệp vụ dàn trải từ thu ngân đến quản lý kho hàng, quản lý khách hàng/đối tác và quản lý tài chính. Để thực hiện được những yêu cầu đó, chúng em đã dành thời gian tìm hiểu, nghiên cứu quy trình bán hàng thực tế ở các cửa hàng bán lẻ và các chuỗi bán hàng thương hiệu lớn như Bách Hoá Xanh, FamilyMart, VinMart+, … Ngoài ra chúng em cũng tham khảo các phần mềm bán hàng hiện có trên thị trường, và kết hợp đọc một số tài liệu về quản lý thu/chi, kiểm kê kho hàng và lập báo cáo tài chính để nắm các yêu cầu nghiệp vụ cơ bản.

Qua quá trình tìm hiểu và xây dựng phần mềm, chúng em đã cơ bản hoàn thành những yêu cầu trên và nắm được quy trình thiết kế một phần mềm. Bên cạnh đó, chúng em cũng học hỏi thêm những kiến thức mới khi tích hợp các thư viện Java vào phần mềm. Tuy nhiên, vì thời gian của môn học giới hạn và sự dàn trải nghiệp vụ đã nêu trên, một số chức năng chúng em thiết kế tượng trưng và để dành sự hoàn thiện đó cho tương lai triển khai phần mềm vào sử dụng thực tế.

Đề tài tuy đã hoàn thành nhưng không tránh khỏi còn những sơ sót, chúng em rất mong nhận được những lời góp ý, chỉ dẫn của quý thầy cô để đề tài hoàn thiện tốt hơn.

Nhóm thực hiện đề tài

**MỤC LỤC**

[**1. PHÂN TÍCH** 1](#_Toc27470470)

[1.1. HIỆN TRẠNG 1](#_Toc27470471)

[1.2. YÊU CẦU HỆ THỐNG 1](#_Toc27470472)

[**2. THIẾT KẾ** 2](#_Toc27470473)

[2.1. MÔ HÌNH TRIỂN KHAI 2](#_Toc27470474)

[2.2. THIẾT KẾ CƠ SỞ DỮ LIỆU 2](#_Toc27470475)

[Chi tiết các bảng 2](#_Toc27470476)

[Sơ đồ quan hệ thực thể 7](#_Toc27470477)

[2.3. THIẾT KẾ GIAO DIỆN 8](#_Toc27470478)

[2.3.1. Sơ đồ tổ chức giao diện 8](#_Toc27470479)

[2.3.2. Thiết kế giao diện cho các chức năng nghiệp vụ 9](#_Toc27470480)

[**3. THỰC HIỆN VIẾT MÃ** 23](#_Toc27470481)

[3. 1. VIẾT MÃ TẠO CSDL 23](#_Toc27470482)

[3.1.1. Tạo CSDL 23](#_Toc27470483)

[3.1.2. SQL truy vấn và thao tác bảng 30](#_Toc27470484)

[3.1.3. Các thủ tục lưu tổng hợp thống kê 38](#_Toc27470485)

[3. 2. LẬP TRÌNH JDBC 42](#_Toc27470486)

[3.2.1. Lớp hỗ trợ 42](#_Toc27470487)

[3.2.2. Model class – Các lớp mô tả dữ liệu 52](#_Toc27470488)

[3.2.3. DAO class – Các lớp truy xuất dữ liệu 95](#_Toc27470489)

# 

# PHÂN TÍCH

## HIỆN TRẠNG

Theo khảo sát tình hình thực tế, có khoảng 85% cửa hàng tạp hoá quy mô vừa và nhỏ vẫn đang được quản lý bằng thao tác theo dõi, ghi chép và tính toán thủ công. Điều này gây khó khăn và tốn kém thời gian cho người quản lý. Nó cũng dẫn đến gia tăng các rủi ro như thất thoát hàng hoá, tiền bạc, kết quả tính toán doanh thu lãi lỗ kém chính xác.

Mặt khác, khi chủ cửa hàng muốn mở rộng quy mô, ngay lập tức đối mặt với các khó khăn về quản lý kho, quản lý nhân viên, quản lý dữ liệu khách hàng và quản lý tài chính. Do đó, rất cần thiết xây dựng giải pháp phần mềm giải quyết những khó khăn trên.

## YÊU CẦU HỆ THỐNG

***“SuperSALES”*** – Phần mềm quản lý siêu thị mini được xây dựng để giải quyết bài toán quản lý trên. Các yêu cầu hệ thống của ***SuperSALES*** như sau:

**Yêu cầu chức năng nghiệp vụ:**

* Mô hình hoá quy trình bán hàng và thanh toán đơn hàng tại chỗ
* Quản lý nhân viên
* Quản lý đối tác: khách hàng, nhà cung cấp sản phẩm
* Quản lý kho hàng: danh sách sản phẩm, xuất nhập hàng, tồn kho, hạn sử dụng
* Quản lý tài chính: quản lý thu chi, quản lý hoá đơn
* Thống kê doanh thu, lãi lỗ, tình trạng kho hàng theo thời gian cụ thể
* Kết xuất các loại báo cáo: doanh thu, nhập xuất hàng, kho hàng

**Yêu cầu bảo mật:**

* Tất cả người dùng phải đăng nhập mới sử dụng được phần mềm
* Admin được phép thực hiện mọi chức năng
* Nhân viên quản lý kho: sử dụng chức năng bán hàng và quản lý kho hàng
* Nhân viên bán hàng: chỉ sử dụng được chức năng bán hàng

**Yêu cầu môi trường công nghệ:**

* Công nghệ Swings và JDBC, chạy trên mọi hệ điều hành với JDK tối thiếu 1.8
* Hệ quản trị CSDL SQL Server 2008 trở lên

# THIẾT KẾ

## MÔ HÌNH TRIỂN KHAI

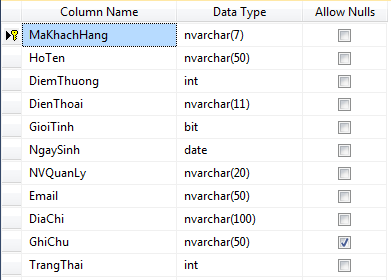
Phần mềm được xây dựng để phục vụ nhiều người dùng nhưng cơ sở dữ liệu thì lưu trữ tập trung.

* Cần một máy cài SQL Server 2008+. Máy này cài hệ điều hành Windows XP trở lên.
* Một máy đặt ở quầy thu ngân để nhân viên sử dụng chức năng bán hàng.

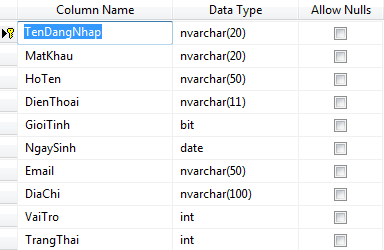
## THIẾT KẾ CƠ SỞ DỮ LIỆU

### Chi tiết các bảng

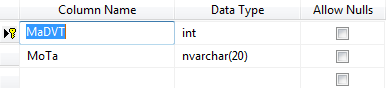
* **Bảng KHACHHANG:**

****

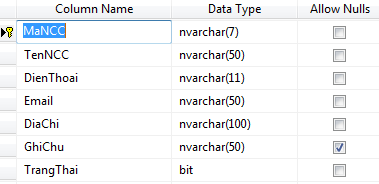
* **Bảng NGUOIDUNG:**

****

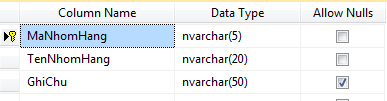
* **Bảng DONVITINH:**

****

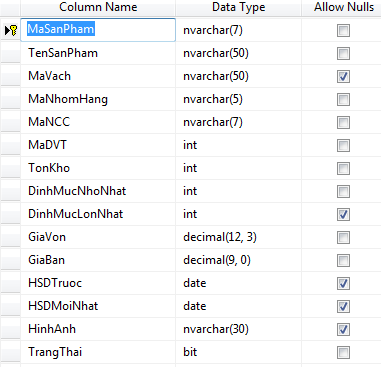
* **Bảng NHACUNGCAP:**

****

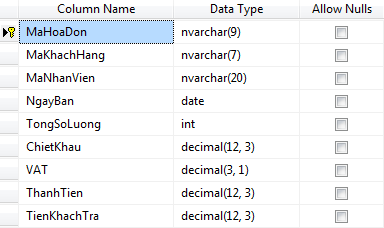
* **Bảng NHOMHANG:**

****

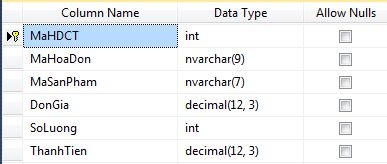
* **Bảng SANPHAM:**

****

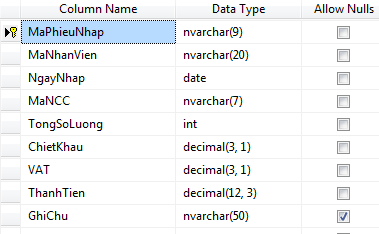
* **Bảng HOADON:**

****

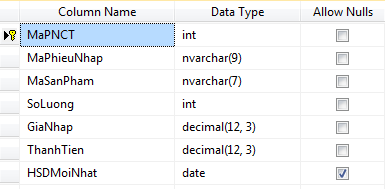
* **Bảng HOADONCHITIET:**

****

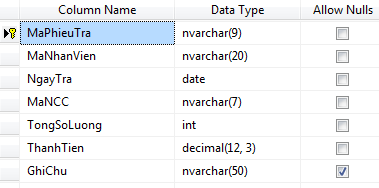
* **Bảng PHIEUNHAPHANG:**

****

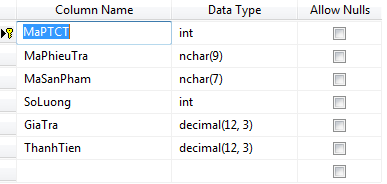
* **Bảng PHIEUNHAPCHITIET:**

****

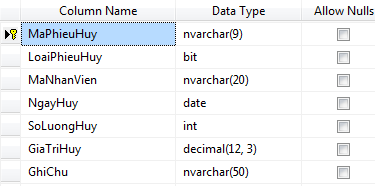
* **Bảng PHIEUTRAHANG:**

****

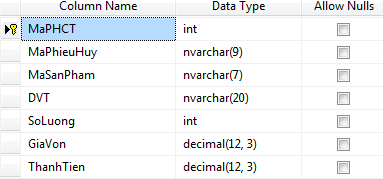
* **Bảng PHIEUTRACHITIET:**

****

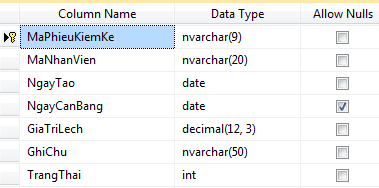
* **Bảng PHIEUHUYHANG:**

****

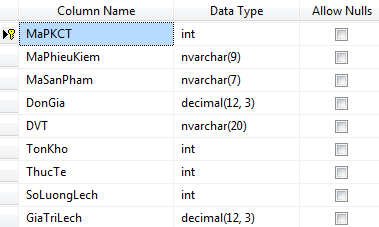
* **Bảng PHIEUHUYCHITIET:**

****

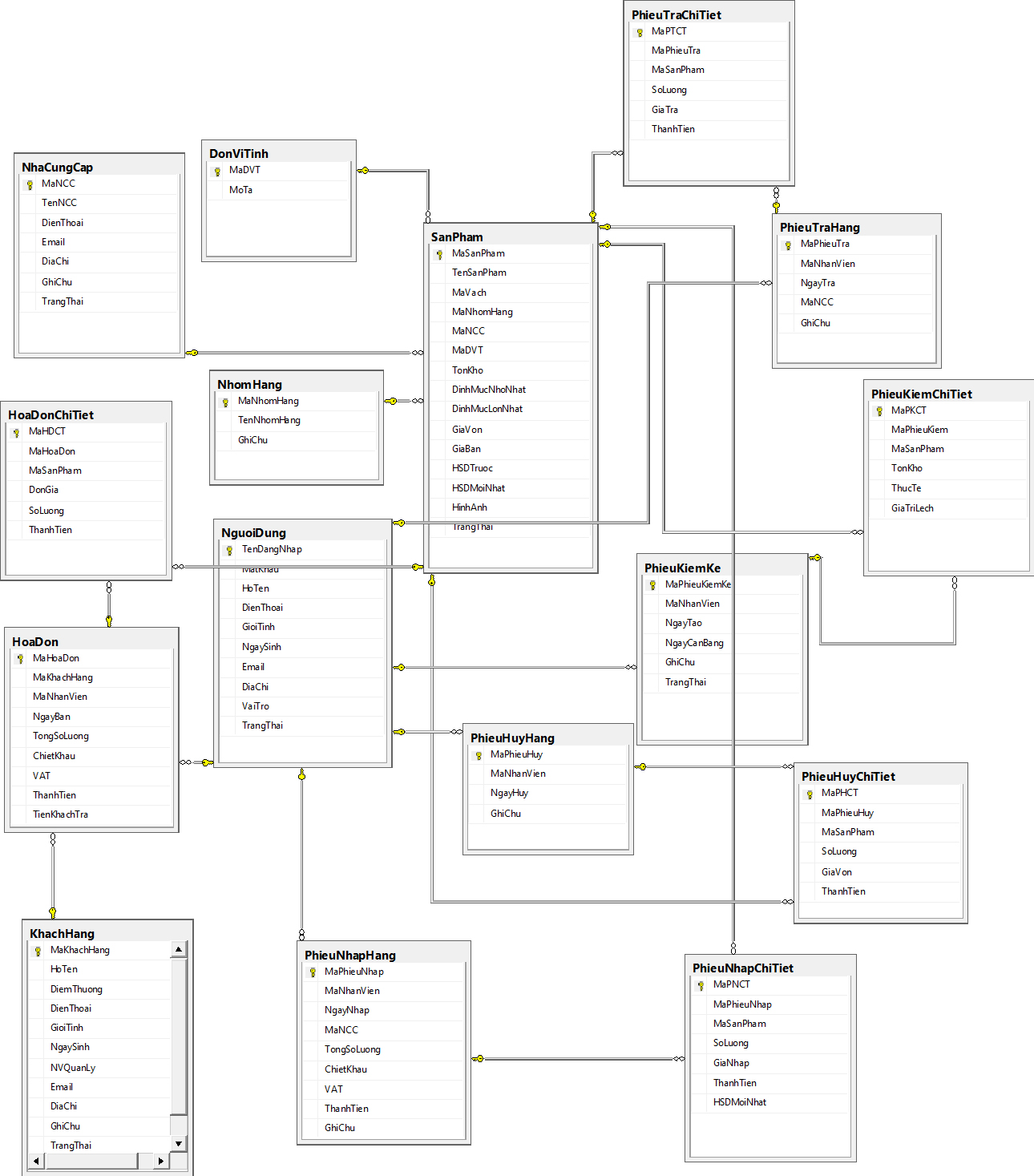
* **Bảng PHIEUKIEMKE:**

****

* **Bảng PHIEUKIEMCHITIET:**

****

### Sơ đồ quan hệ thực thể



## THIẾT KẾ GIAO DIỆN

### Sơ đồ tổ chức giao diện

Màn hình chào

Đăng nhập

Giao diện bán hàng

Giao diện quản lý

Thêm Khách Hàng

Thông tin tài khoản

Quản lý Tài Khoản

Quản lý Nhân Viên

Quản lý Thu Chi

Quản lý Đối Tác

Báo cáo/Thống kê

Quản lý Giao Dịch

Quản lý Sản Phẩm

Quản lý Khách Hàng

Quản lý Nhà Cung Cấp

Kiểm kê

Quản lý Hoá Đơn

Quản lý Xuất Huỷ Hàng

Quản lý Trả Hàng

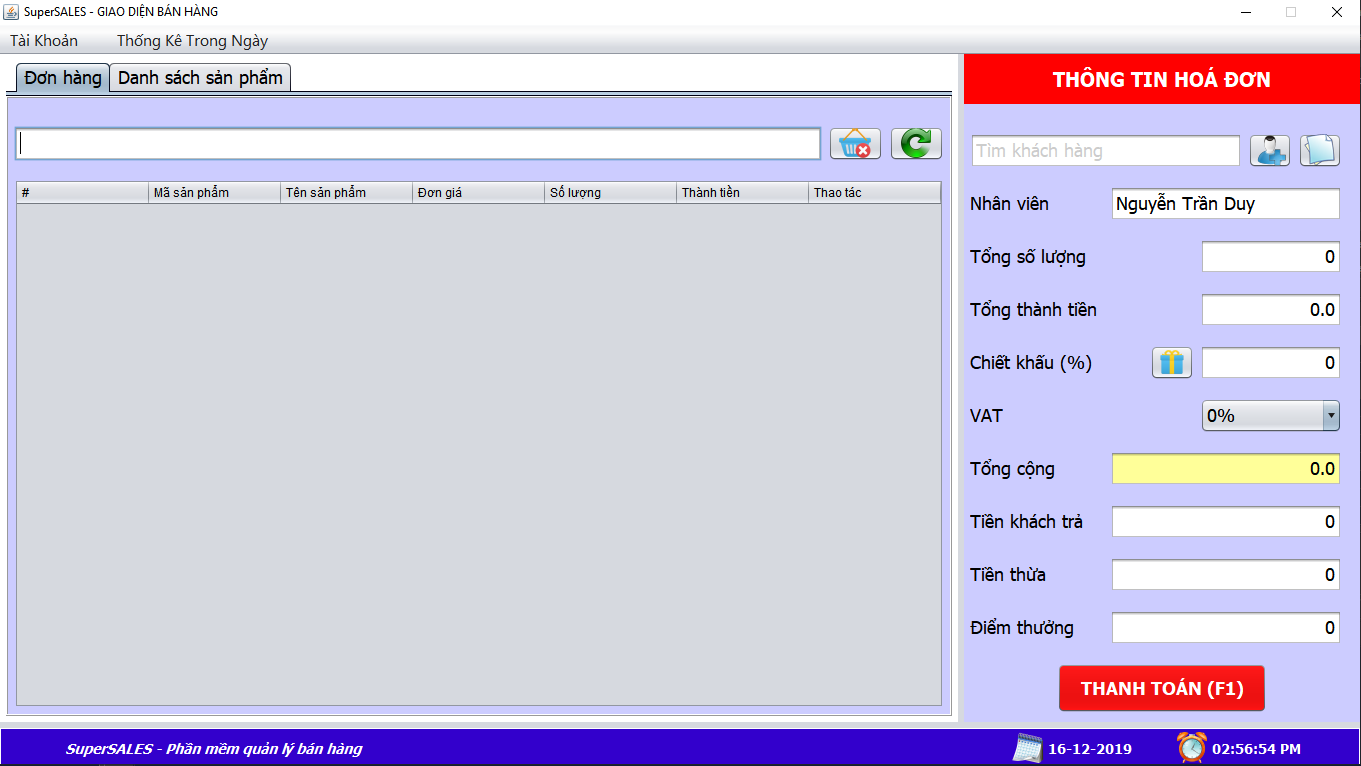
Quản lý Nhập Hàng

### Thiết kế giao diện cho các chức năng nghiệp vụ

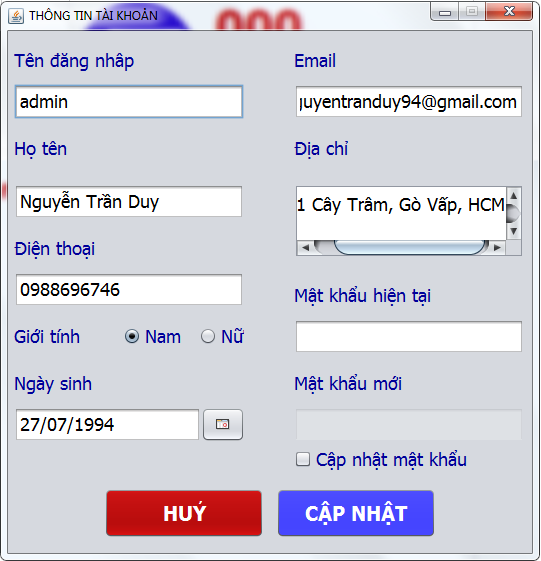
* + - 1. **Màn hình đăng nhập**



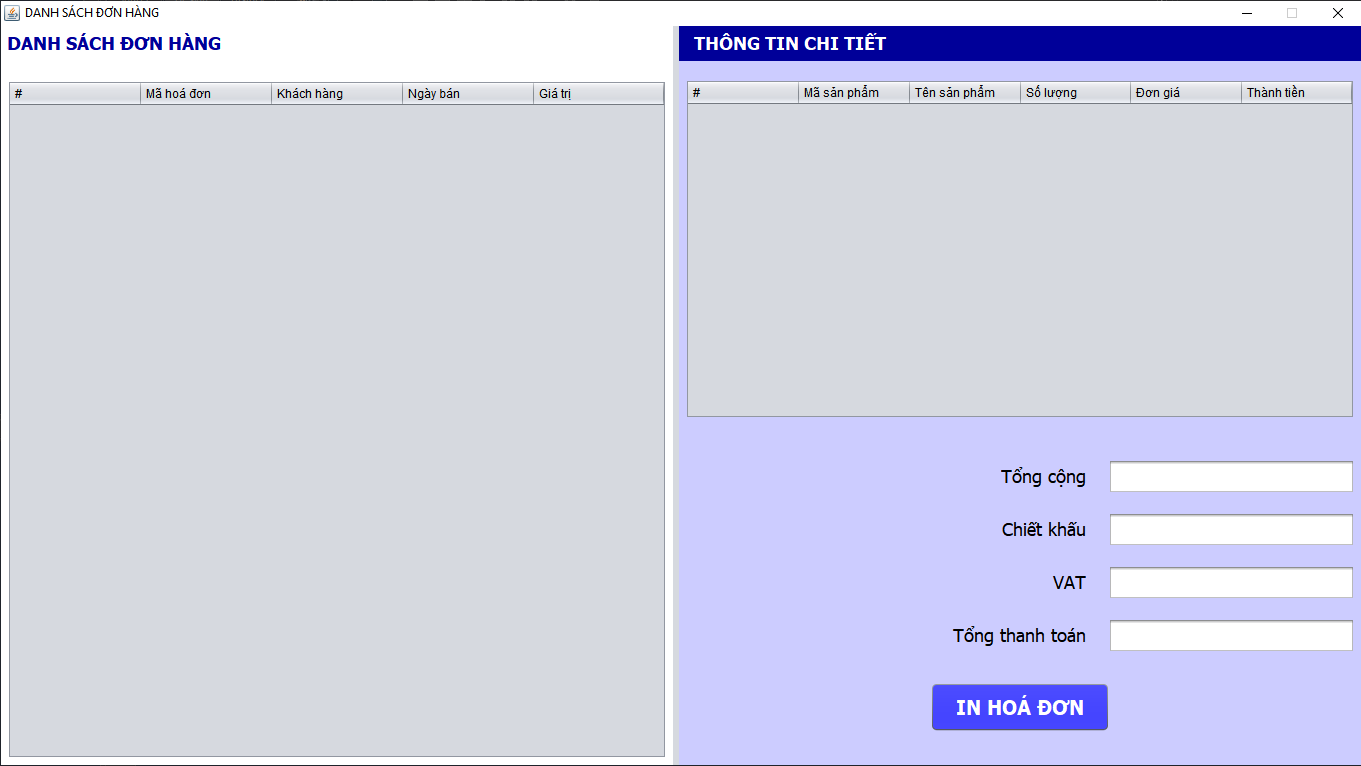
* + - 1. **Giao diện bán hàng**



* + - 1. **Giao diện Thông tin tài khoản**



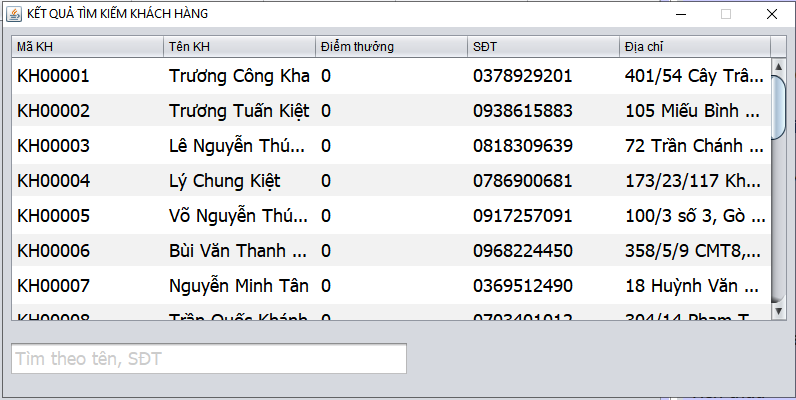
* + - 1. **Giao diện Danh Sách Đơn Hàng (trong Bán Hàng)**



* + - 1. **Giao diện Tổng Kết Cuối Ngày (trong Bán Hàng)**



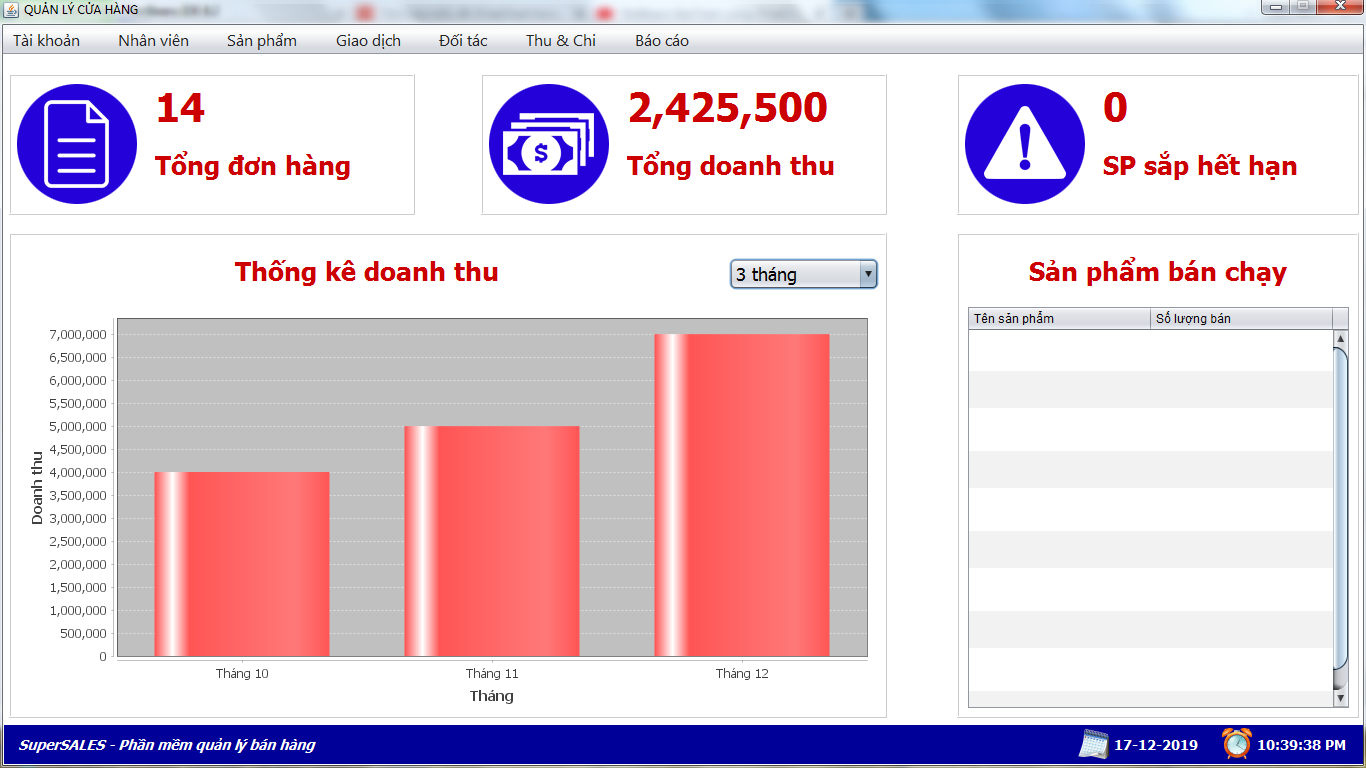
* + - 1. **Giao diện Tìm Khách Hàng (trong Bán Hàng)**



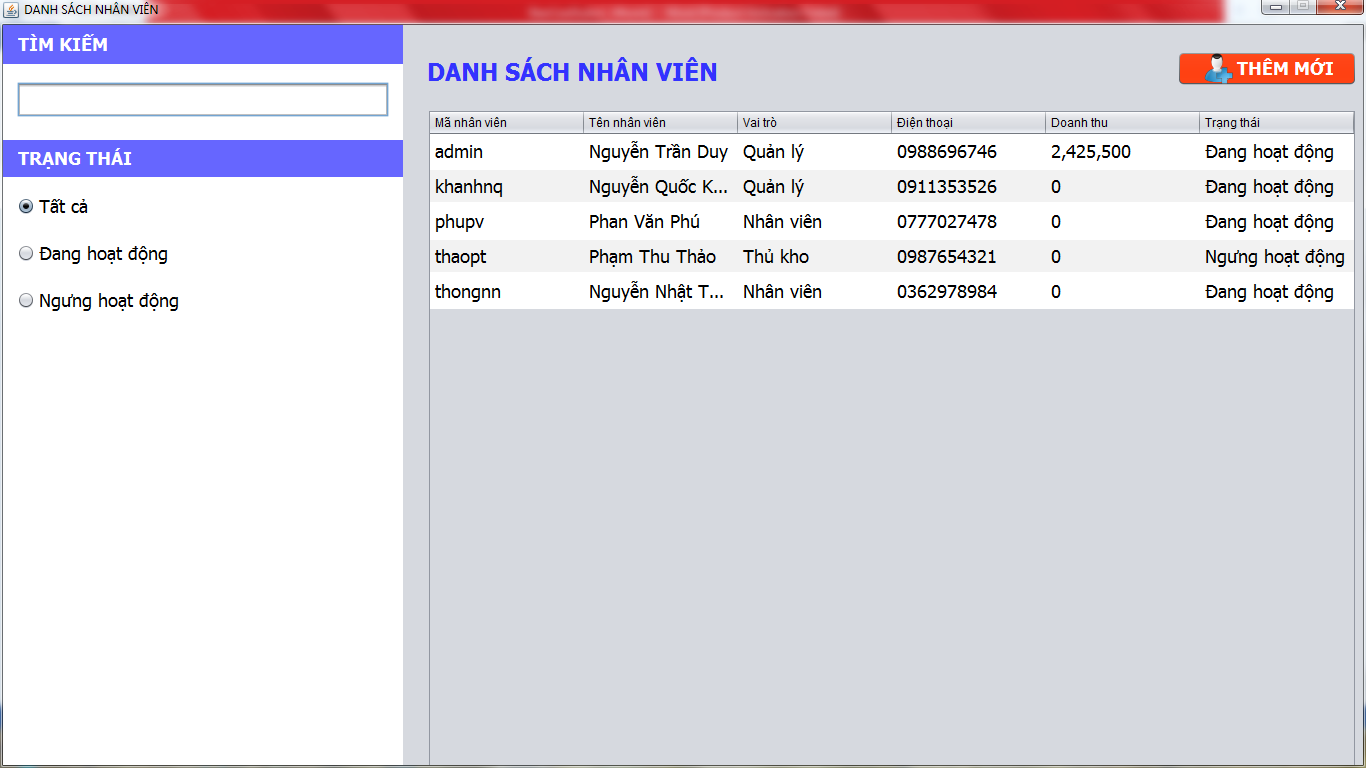
* + - 1. **Giao diện Chiết Khấu (trong Bán Hàng)**



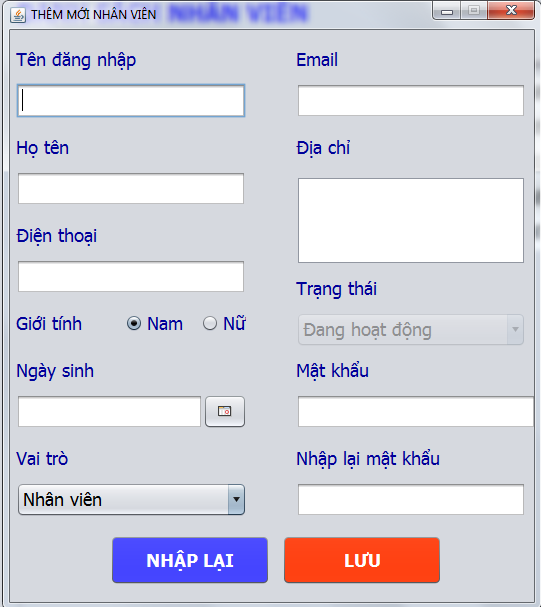
* + - 1. **Giao diện Quản Lý**



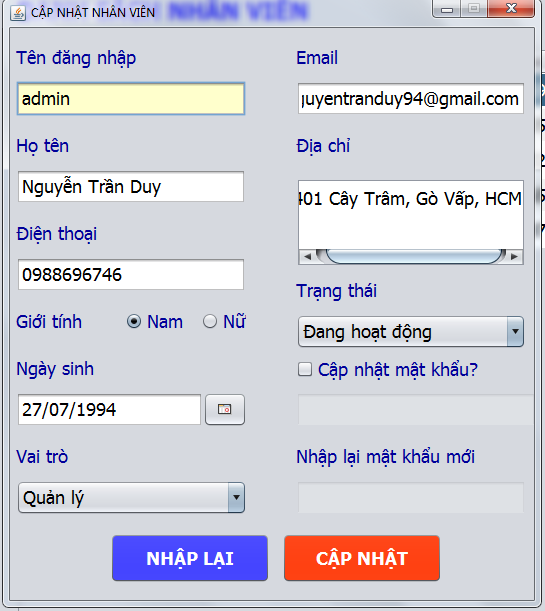
* + - 1. **Giao diện Quản Lý Nhân Viên**



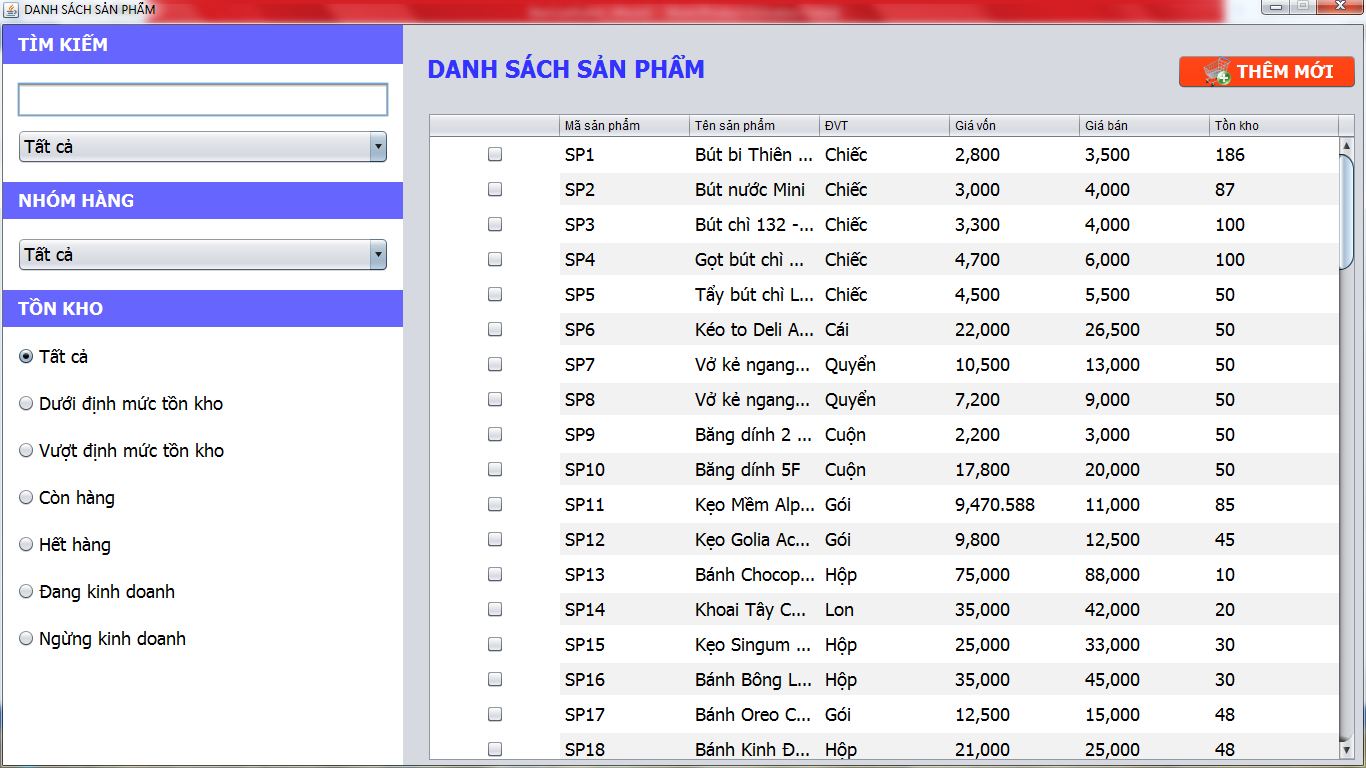
* + - 1. **Giao diện Thêm Mới Nhân Viên**



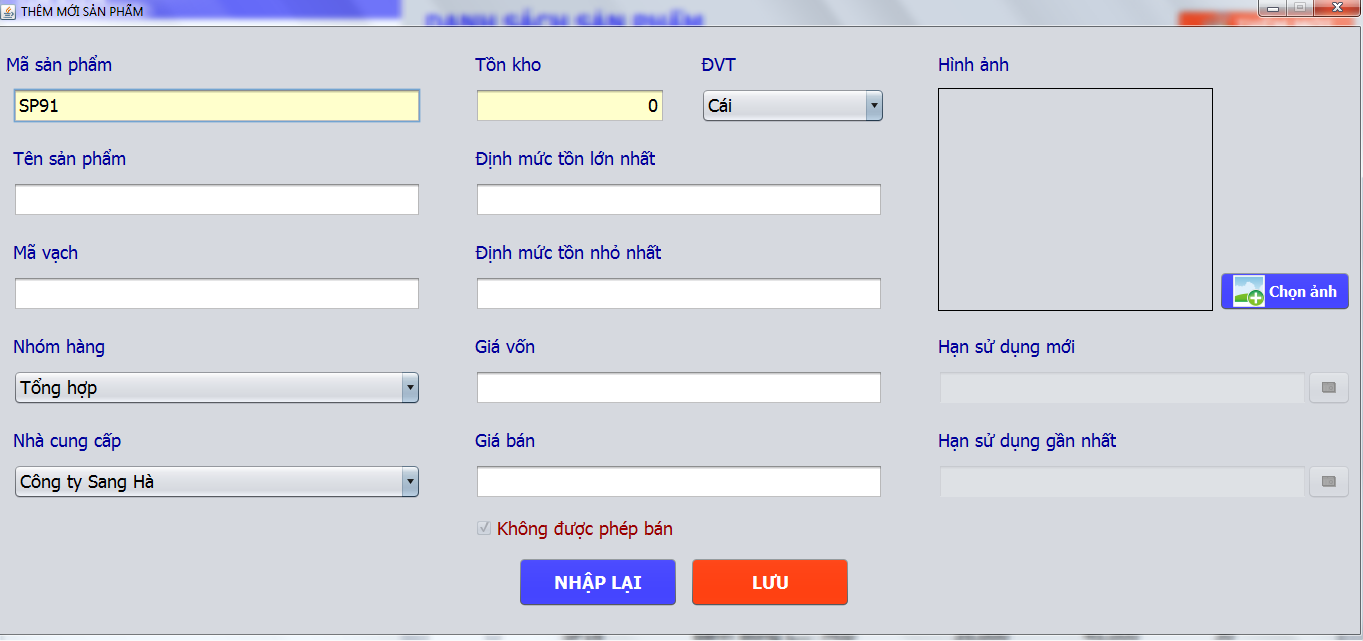
* + - 1. **Giao diện Cập Nhật Nhân Viên**



* + - 1. **Giao diện Quản Lý Sản Phẩm**



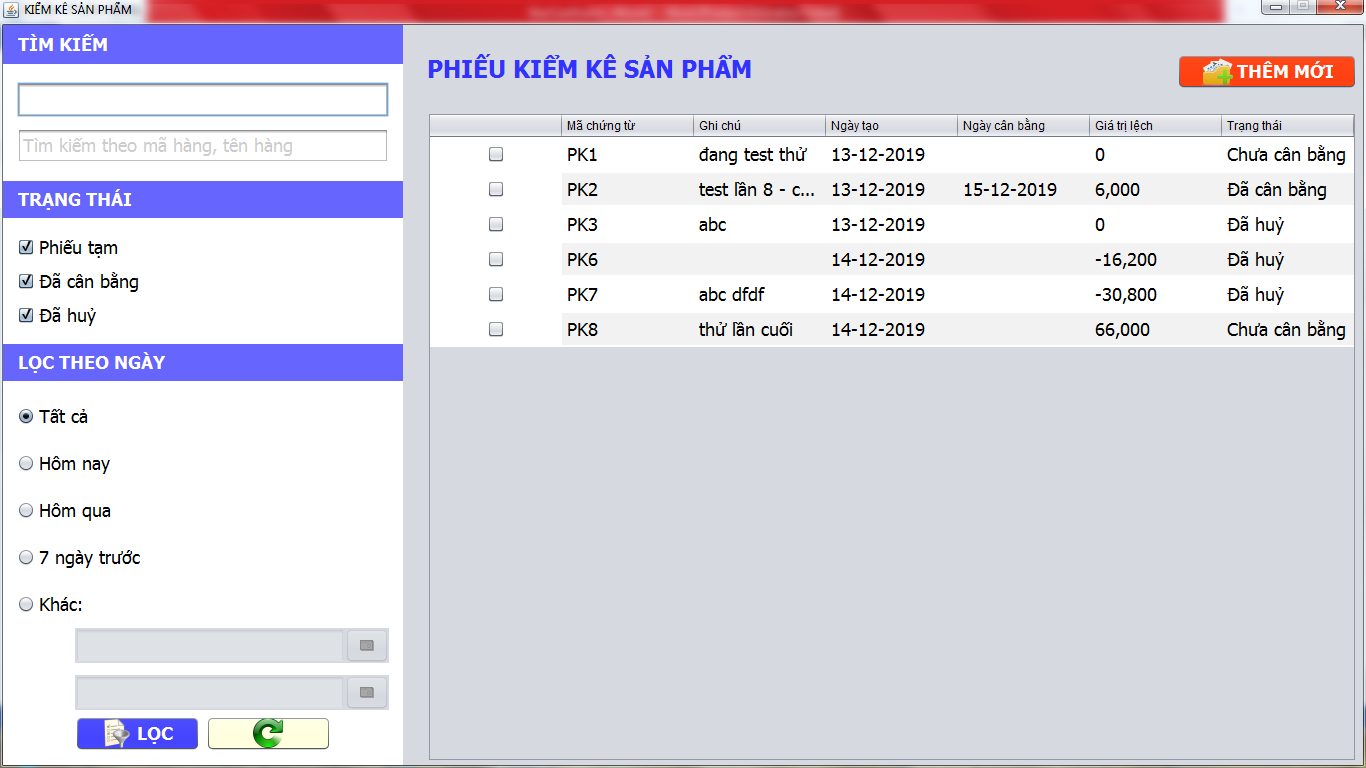
* + - 1. **Giao diện Thêm Mới Sản Phẩm**



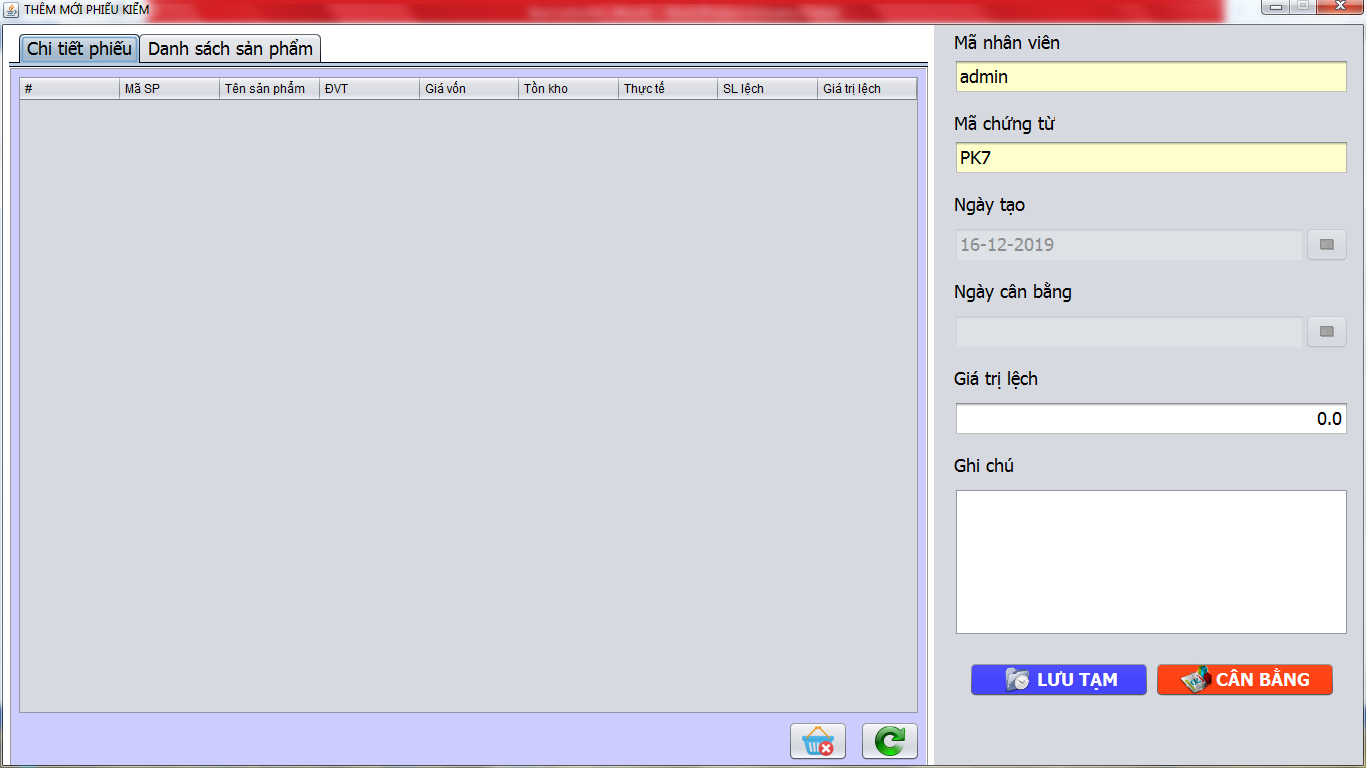
* + - 1. **Giao diện Cập Nhật Sản Phẩm**



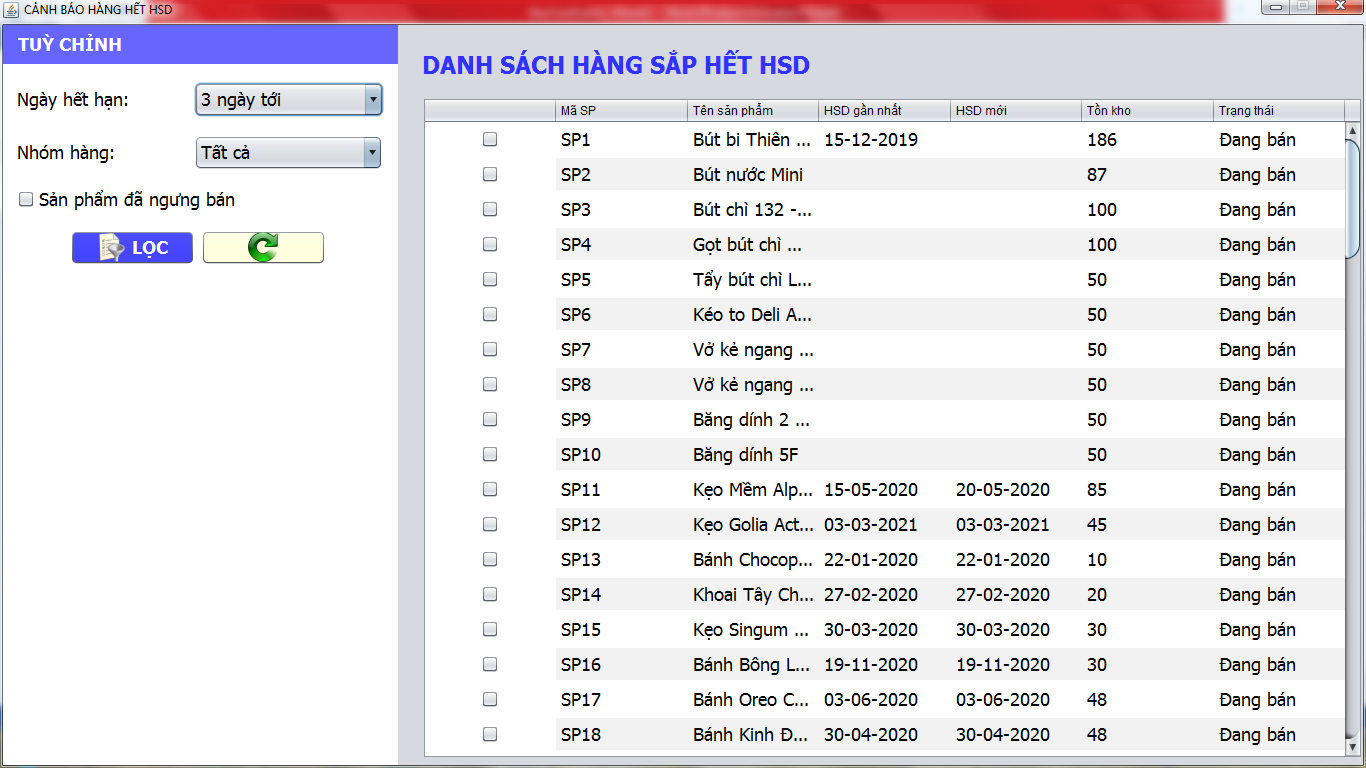
* + - 1. **Giao diện Kiểm Kê Sản Phẩm**



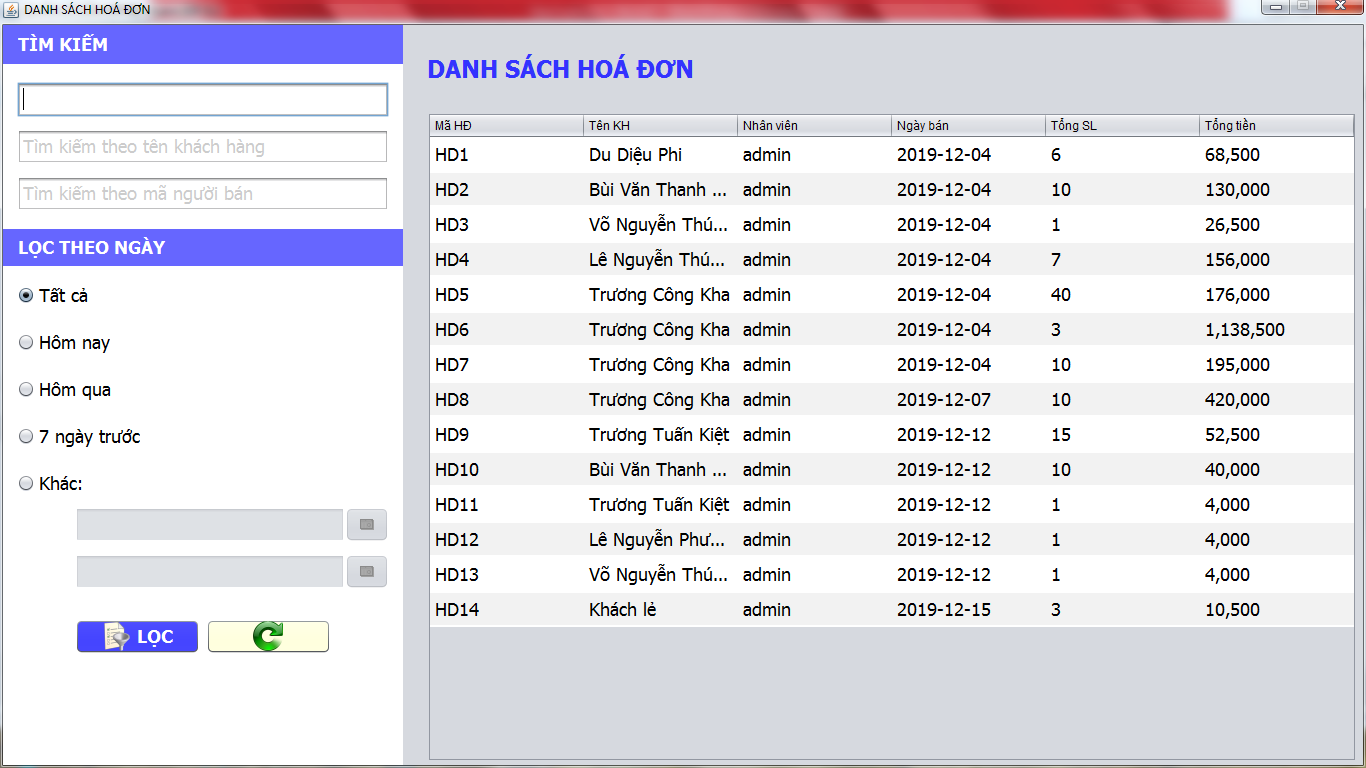
* + - 1. **Giao diện Thêm Mới Phiếu Kiểm Kê**



* + - 1. **Giao diện Cảnh Báo Sản Phẩm Hết Hạn Sử Dụng**



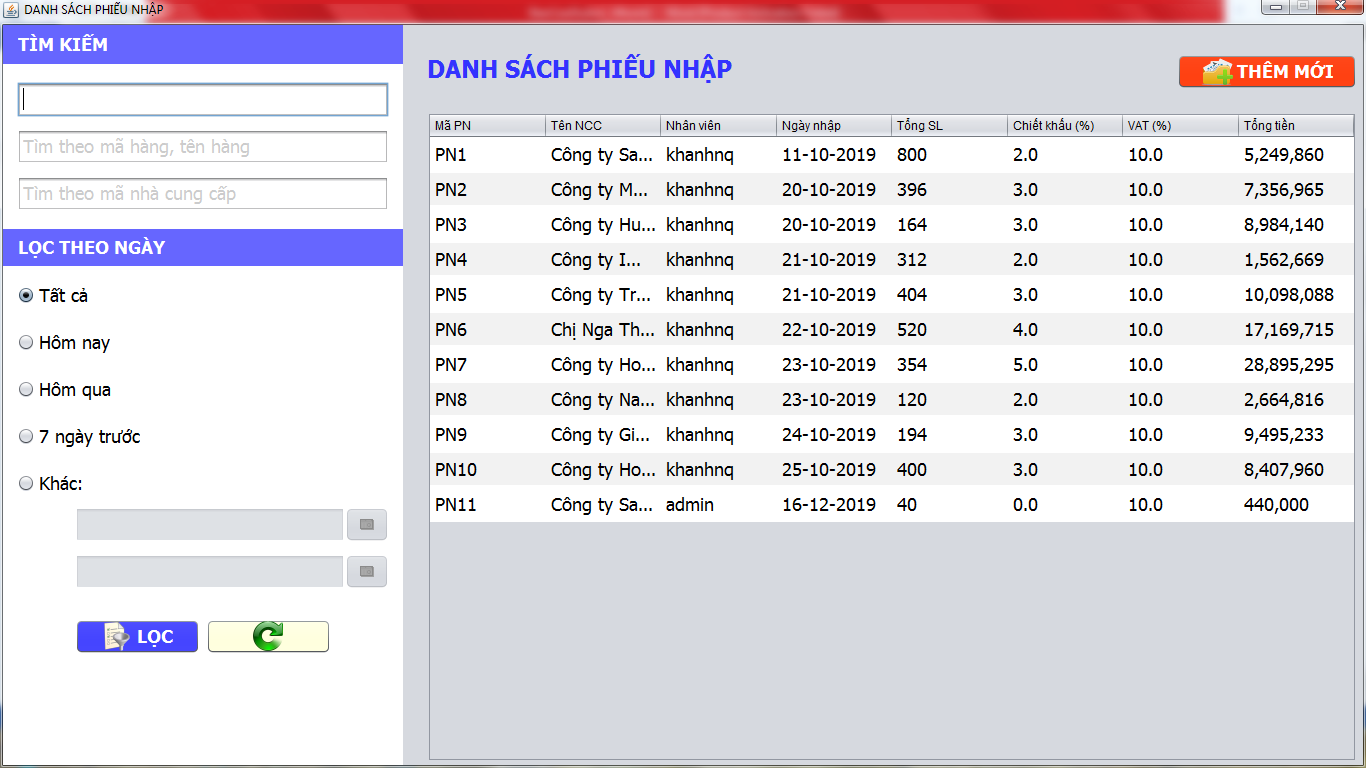
* + - 1. **Giao diện Quản Lý Hoá Đơn**



* + - 1. **Giao diện Quản Lý Chi Tiết Hoá Đơn**



* + - 1. **Giao diện Nhập Hàng**



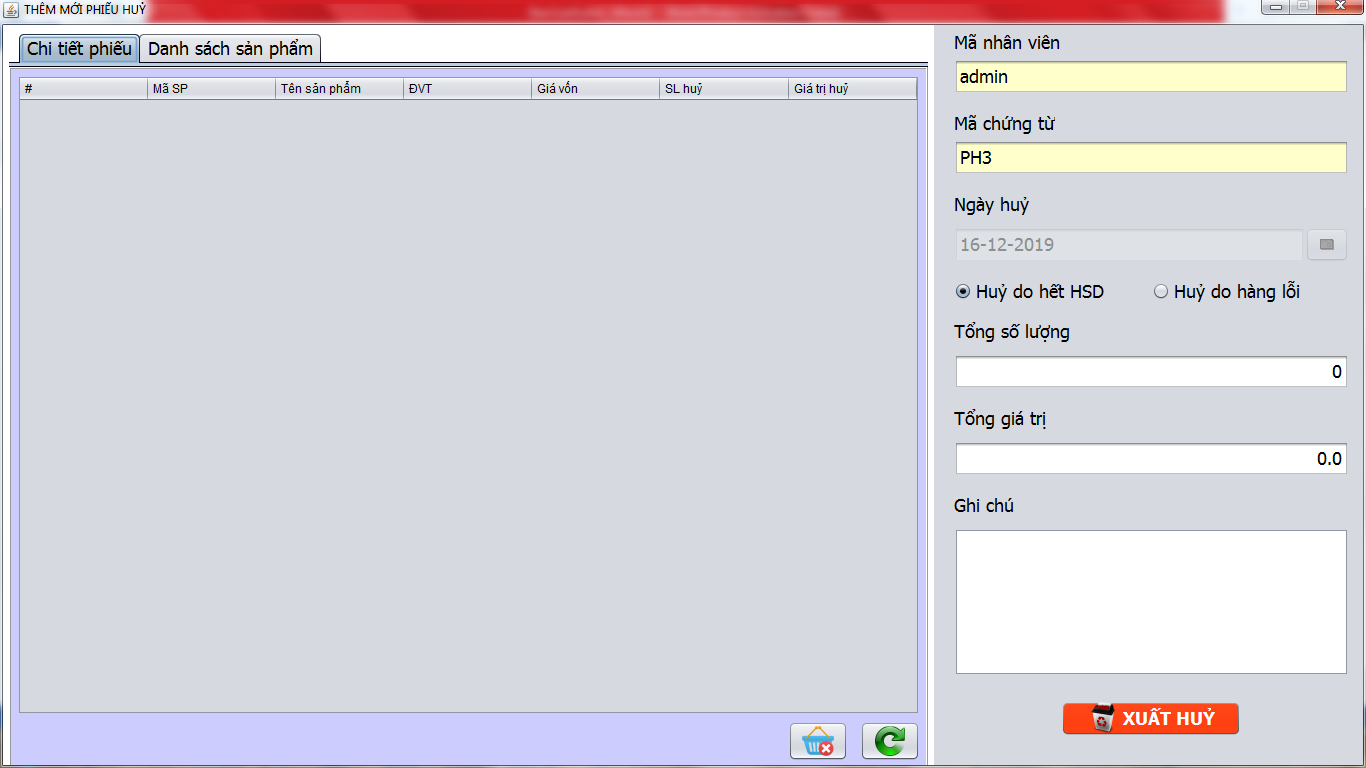
* + - 1. **Giao diện Thêm Mới Phiếu Nhập Hàng**



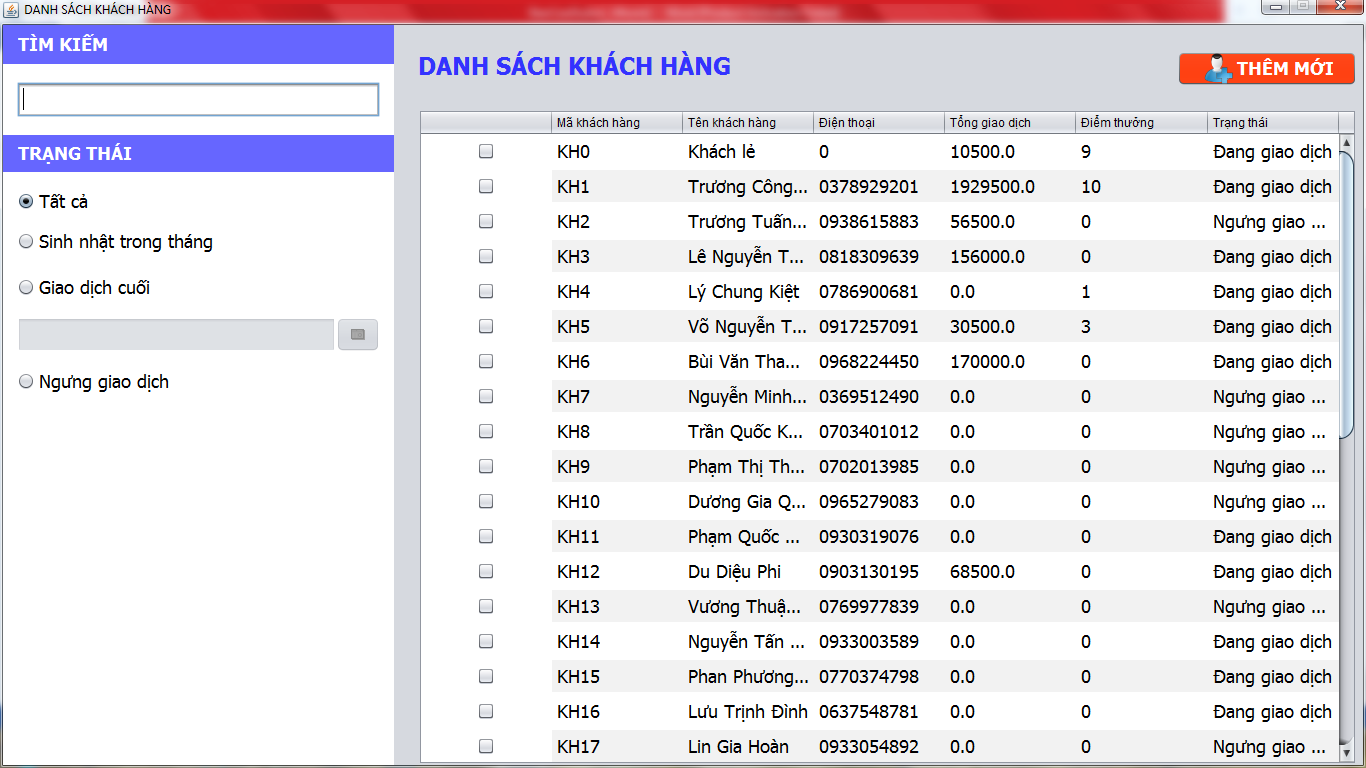
* + - 1. **Giao diện Xuất Huỷ Hàng**



* + - 1. **Giao diện Thêm Mới Phiếu Xuất Huỷ**



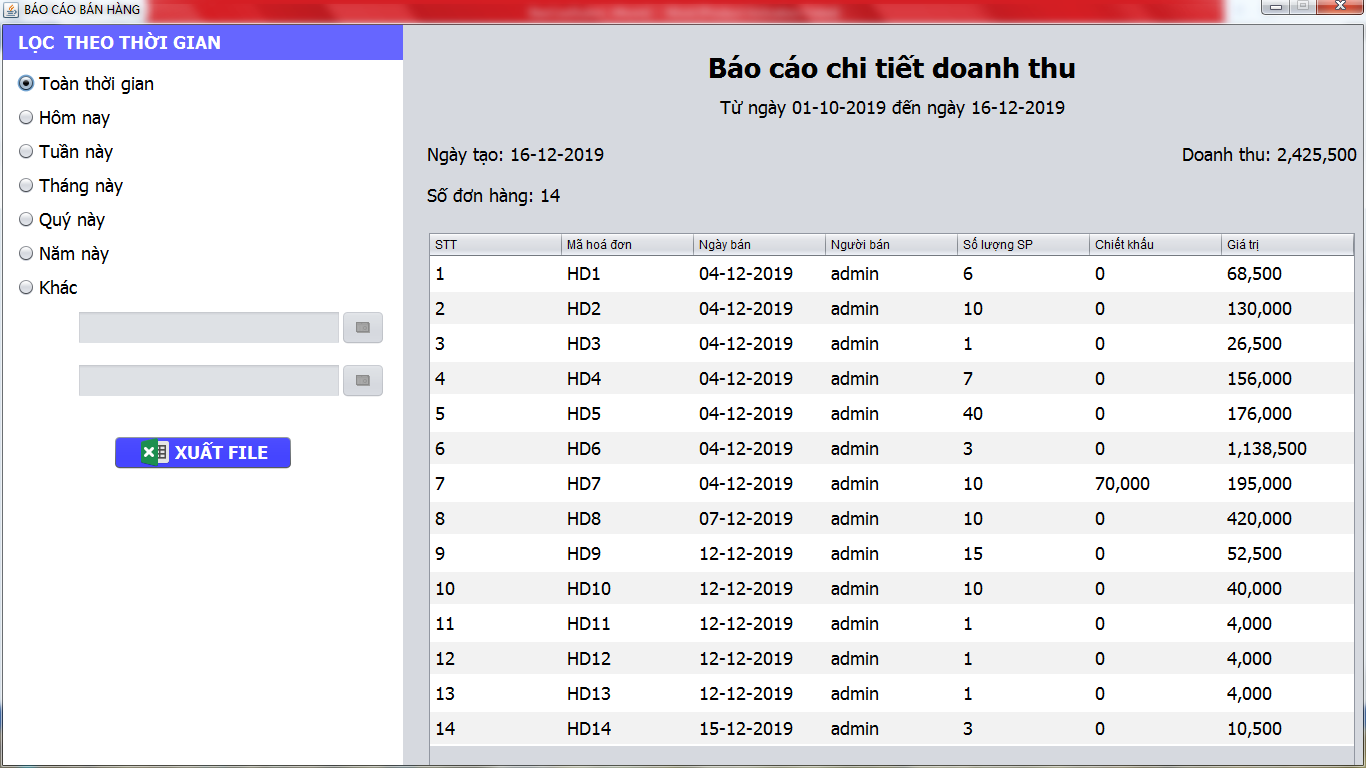
* + - 1. **Giao diện Quản Lý Khách Hàng**



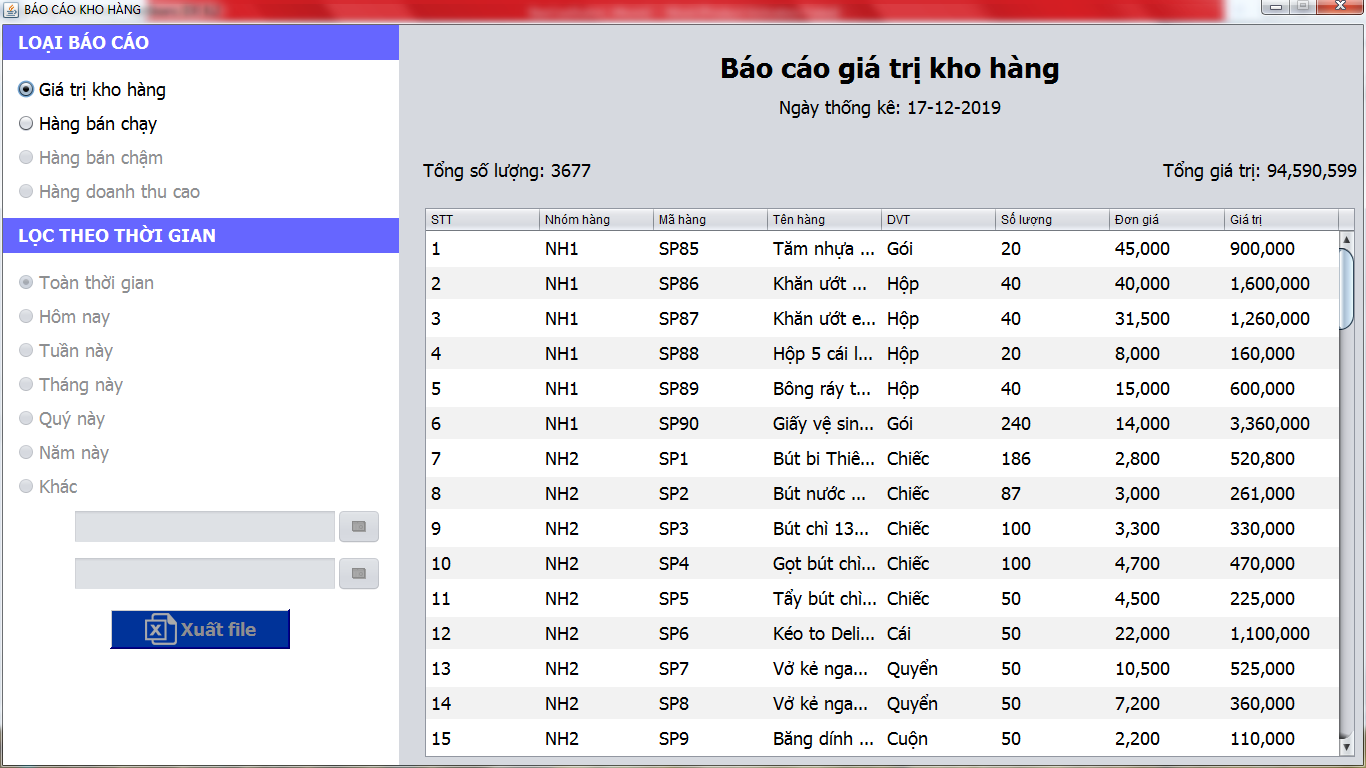
* + - 1. **Giao diện Quản Lý Nhà Cung Cấp**



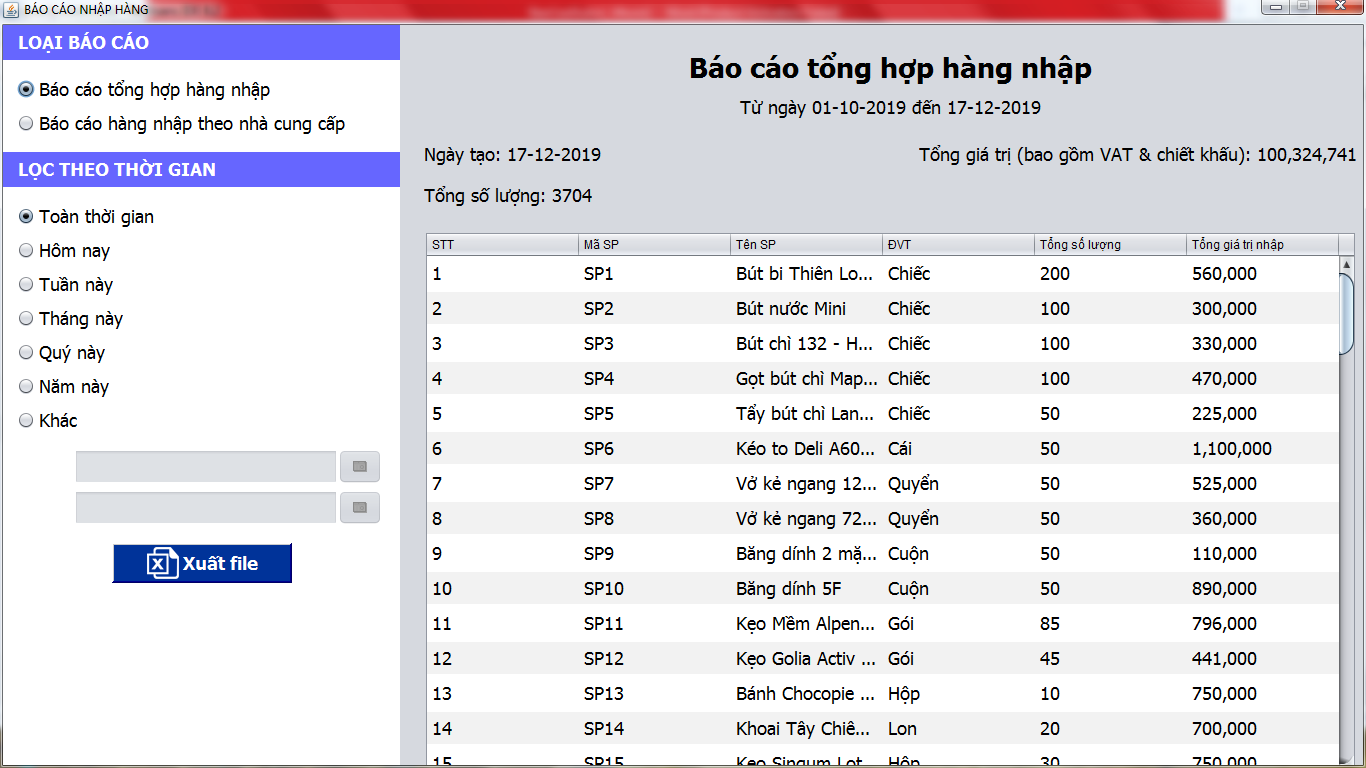
* + - 1. **Giao diện Báo Cáo Bán Hàng**



* + - 1. **Giao diện Báo Cáo Kho Hàng**



* + - 1. **Giao diện Báo Cáo Nhập Hàng**



* + - 1. **Giao diện Báo Cáo Tài Chính**



# THỰC HIỆN VIẾT MÃ

## VIẾT MÃ TẠO CSDL

### Tạo CSDL

USE [master]

GO

CREATE DATABASE [quanlysieuthimini]

GO

USE [quanlysieuthimini]

GO

CREATE TABLE [dbo].[DonViTinh](

[MaDVT] [int] IDENTITY(1,1) NOT NULL,

[MoTa] [nvarchar](20) NOT NULL,

CONSTRAINT [PK\_DonViTinh] PRIMARY KEY CLUSTERED

(

[MaDVT] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[HoaDon](

[MaHoaDon] [nvarchar](9) NOT NULL,

[MaKhachHang] [nvarchar](7) NOT NULL,

[MaNhanVien] [nvarchar](20) NOT NULL,

[NgayBan] [date] NOT NULL,

[TongSoLuong] [int] NOT NULL,

[ChietKhau] [decimal](12, 3) NOT NULL,

[VAT] [decimal](3, 1) NOT NULL,

[ThanhTien] [decimal](12, 3) NOT NULL,

[TienKhachTra] [decimal](12, 3) NOT NULL,

CONSTRAINT [PK\_HoaDon] PRIMARY KEY CLUSTERED

(

[MaHoaDon] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[HoaDonChiTiet](

[MaHDCT] [int] IDENTITY(1,1) NOT NULL,

[MaHoaDon] [nvarchar](9) NOT NULL,

[MaSanPham] [nvarchar](7) NOT NULL,

[DonGia] [decimal](12, 3) NOT NULL,

[SoLuong] [int] NOT NULL,

[ThanhTien] [decimal](12, 3) NOT NULL,

CONSTRAINT [PK\_HoaDonChiTiet] PRIMARY KEY CLUSTERED

(

[MaHDCT] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[KhachHang](

[MaKhachHang] [nvarchar](7) NOT NULL,

[HoTen] [nvarchar](50) NOT NULL,

[DiemThuong] [int] NOT NULL,

[DienThoai] [nvarchar](11) NOT NULL,

[GioiTinh] [bit] NOT NULL,

[NgaySinh] [date] NOT NULL,

[NVQuanLy] [nvarchar](20) NOT NULL,

[Email] [nvarchar](50) NOT NULL,

[DiaChi] [nvarchar](100) NOT NULL,

[GhiChu] [nvarchar](50) NULL,

[TrangThai] [int] NOT NULL,

CONSTRAINT [PK\_KhachHang] PRIMARY KEY CLUSTERED

(

[MaKhachHang] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[NguoiDung](

[TenDangNhap] [nvarchar](20) NOT NULL,

[MatKhau] [nvarchar](20) NOT NULL,

[HoTen] [nvarchar](50) NOT NULL,

[DienThoai] [nvarchar](11) NOT NULL,

[GioiTinh] [bit] NOT NULL,

[NgaySinh] [date] NOT NULL,

[Email] [nvarchar](50) NOT NULL,

[DiaChi] [nvarchar](100) NOT NULL,

[VaiTro] [int] NOT NULL,

[TrangThai] [int] NOT NULL,

CONSTRAINT [PK\_NguoiDung] PRIMARY KEY CLUSTERED

(

[TenDangNhap] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[NhaCungCap](

[MaNCC] [nvarchar](7) NOT NULL,

[TenNCC] [nvarchar](50) NOT NULL,

[DienThoai] [nvarchar](11) NOT NULL,

[Email] [nvarchar](50) NOT NULL,

[DiaChi] [nvarchar](100) NOT NULL,

[GhiChu] [nvarchar](50) NULL,

[TrangThai] [bit] NOT NULL,

CONSTRAINT [PK\_NhaCungCap] PRIMARY KEY CLUSTERED

(

[MaNCC] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[NhomHang](

[MaNhomHang] [nvarchar](5) NOT NULL,

[TenNhomHang] [nvarchar](20) NOT NULL,

[GhiChu] [nvarchar](50) NULL,

CONSTRAINT [PK\_NhomHang] PRIMARY KEY CLUSTERED

(

[MaNhomHang] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[PhieuHuyChiTiet](

[MaPHCT] [int] IDENTITY(1,1) NOT NULL,

[MaPhieuHuy] [nvarchar](9) NOT NULL,

[MaSanPham] [nvarchar](7) NOT NULL,

[DVT] [nvarchar](20) NOT NULL,

[SoLuong] [int] NOT NULL,

[GiaVon] [decimal](12, 3) NOT NULL,

[ThanhTien] [decimal](12, 3) NOT NULL,

CONSTRAINT [PK\_PhieuHuyChiTiet] PRIMARY KEY CLUSTERED

(

[MaPHCT] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[PhieuHuyHang](

[MaPhieuHuy] [nvarchar](9) NOT NULL,

[LoaiPhieuHuy] [bit] NOT NULL,

[MaNhanVien] [nvarchar](20) NOT NULL,

[NgayHuy] [date] NOT NULL,

[SoLuongHuy] [int] NOT NULL,

[GiaTriHuy] [decimal](12, 3) NOT NULL,

[GhiChu] [nvarchar](50) NOT NULL,

CONSTRAINT [PK\_PhieuHuyHang] PRIMARY KEY CLUSTERED

(

[MaPhieuHuy] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[PhieuKiemChiTiet](

[MaPKCT] [int] IDENTITY(1,1) NOT NULL,

[MaPhieuKiem] [nvarchar](9) NOT NULL,

[MaSanPham] [nvarchar](7) NOT NULL,

[DonGia] [decimal](12, 3) NOT NULL,

[DVT] [nvarchar](20) NOT NULL,

[TonKho] [int] NOT NULL,

[ThucTe] [int] NOT NULL,

[SoLuongLech] [int] NOT NULL,

[GiaTriLech] [decimal](12, 3) NOT NULL,

CONSTRAINT [PK\_PhieuKiemChiTiet] PRIMARY KEY CLUSTERED

(

[MaPKCT] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[PhieuKiemKe](

[MaPhieuKiemKe] [nvarchar](9) NOT NULL,

[MaNhanVien] [nvarchar](20) NOT NULL,

[NgayTao] [date] NOT NULL,

[NgayCanBang] [date] NULL,

[GiaTriLech] [decimal](12, 3) NOT NULL,

[GhiChu] [nvarchar](50) NOT NULL,

[TrangThai] [int] NOT NULL,

CONSTRAINT [PK\_PhieuKiemKe] PRIMARY KEY CLUSTERED

(

[MaPhieuKiemKe] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[PhieuNhapChiTiet](

[MaPNCT] [int] IDENTITY(1,1) NOT NULL,

[MaPhieuNhap] [nvarchar](9) NOT NULL,

[MaSanPham] [nvarchar](7) NOT NULL,

[SoLuong] [int] NOT NULL,

[GiaNhap] [decimal](12, 3) NOT NULL,

[ThanhTien] [decimal](12, 3) NOT NULL,

[HSDMoiNhat] [date] NULL,

CONSTRAINT [PK\_PhieuNhapChiTiet] PRIMARY KEY CLUSTERED

(

[MaPNCT] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[PhieuNhapHang](

[MaPhieuNhap] [nvarchar](9) NOT NULL,

[MaNhanVien] [nvarchar](20) NOT NULL,

[NgayNhap] [date] NOT NULL,

[MaNCC] [nvarchar](7) NOT NULL,

[TongSoLuong] [int] NOT NULL,

[ChietKhau] [decimal](3, 1) NOT NULL,

[VAT] [decimal](3, 1) NOT NULL,

[ThanhTien] [decimal](12, 3) NOT NULL,

[GhiChu] [nvarchar](50) NULL,

CONSTRAINT [PK\_PhieuNhapHang] PRIMARY KEY CLUSTERED

(

[MaPhieuNhap] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[PhieuTraChiTiet](

[MaPTCT] [int] NOT NULL,

[MaPhieuTra] [nvarchar](9) NOT NULL,

[MaSanPham] [nvarchar](7) NOT NULL,

[SoLuong] [int] NOT NULL,

[GiaTra] [decimal](12, 3) NOT NULL,

[ThanhTien] [decimal](12, 3) NOT NULL,

CONSTRAINT [PK\_PhieuTraChiTiet] PRIMARY KEY CLUSTERED

(

[MaPTCT] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[PhieuTraHang](

[MaPhieuTra] [nvarchar](9) NOT NULL,

[MaNhanVien] [nvarchar](20) NOT NULL,

[NgayTra] [date] NOT NULL,

[MaNCC] [nvarchar](7) NOT NULL,

[TongSoLuong] [int] NOT NULL,

[ThanhTien] [decimal](12, 3) NOT NULL,

[GhiChu] [nvarchar](50) NULL,

CONSTRAINT [PK\_PhieuTraHang] PRIMARY KEY CLUSTERED

(

[MaPhieuTra] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

CREATE TABLE [dbo].[SanPham](

[MaSanPham] [nvarchar](7) NOT NULL,

[TenSanPham] [nvarchar](50) NOT NULL,

[MaVach] [nvarchar](50) NULL,

[MaNhomHang] [nvarchar](5) NOT NULL,

[MaNCC] [nvarchar](7) NOT NULL,

[MaDVT] [int] NOT NULL,

[TonKho] [int] NOT NULL,

[DinhMucNhoNhat] [int] NOT NULL,

[DinhMucLonNhat] [int] NULL,

[GiaVon] [decimal](12, 3) NOT NULL,

[GiaBan] [decimal](9, 0) NOT NULL,

[HSDTruoc] [date] NULL,

[HSDMoiNhat] [date] NULL,

[HinhAnh] [nvarchar](30) NULL,

[TrangThai] [bit] NOT NULL,

CONSTRAINT [PK\_SanPham] PRIMARY KEY CLUSTERED

(

[MaSanPham] ASC

)WITH (PAD\_INDEX = OFF, STATISTICS\_NORECOMPUTE = OFF, IGNORE\_DUP\_KEY = OFF, ALLOW\_ROW\_LOCKS = ON, ALLOW\_PAGE\_LOCKS = ON) ON [PRIMARY]

) ON [PRIMARY]

GO

### SQL truy vấn và thao tác bảng

* + - 1. **Bảng DonViTinh**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn tất cả | SELECT \* FROM DonViTinh |
| Truy vấn theo mã | SELECT \* FROM DonViTinh WHERE MaDVT = ? |

* + - 1. **Bảng NguoiDung**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn tất cả | SELECT \* FROM NguoiDung |
| Truy vấn theo tên đăng nhập | SELECT \* FROM NguoiDung WHERE TenDangNhap = ? |
| Truy vấn theo email | SELECT \* FROM NguoiDung WHERE Email = ? |
| Truy vấn theo SĐT | SELECT \* FROM NguoiDung WHERE DienThoai = ? |
| Truy vấn theo tên hoặc mã hoặc SĐT | SELECT \* FROM NguoiDung WHERE  TenDangNhap LIKE ?  OR HoTen LIKE ?  OR DienThoai LIKE ? |
| Truy vấn theo trạng thái | SELECT \* FROM NguoiDung WHERE TrangThai = ? |
| Thêm mới người dùng | INSERT INTO NguoiDung(TenDangNhap, MatKhau, HoTen, DienThoai, GioiTinh, NgaySinh, Email, DiaChi, VaiTro, TrangThai)  VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?) |
| Cập nhật người dùng | UPDATE NguoiDung SET  MatKhau = ?, HoTen = ?, DienThoai = ?, GioiTinh = ?, NgaySinh = ?, Email = ?, DiaChi = ?, VaiTro = ?, TrangThai = ?  WHERE TenDangNhap = ? |

* + - 1. **Bảng KhachHang**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn tất cả và sắp xếp theo mã khách hàng | SELECT \* FROM KhachHang  ORDER BY CAST(RIGHT(MaKhachHang,LEN(MaKhachHang)-2) AS int) |
| Truy vấn theo mã | SELECT \* FROM KhachHang WHERE MaKhachHang = ? |
| Truy vấn theo email | SELECT \* FROM KhachHang WHERE Email LIKE ? |
| Truy vấn theo SĐT | SELECT \* FROM KhachHang WHERE DienThoai = ? |
| Truy vấn theo tên hoặc mã hoặc SĐT | SELECT \* FROM KhachHang WHERE  HoTen LIKE ?  OR MaKhachHang LIKE ?  OR DienThoai LIKE ? |
| Truy vấn theo trạng thái | SELECT \* FROM KhachHang WHERE TrangThai = ? |
| Truy vấn theo tháng sinh | SELECT \* FROM KhachHang WHERE  MONTH(NgaySinh) = ? |
| Truy vấn theo giao dịch cuối | SELECT \* FROM KhachHang WHERE NOT EXISTS  (SELECT \* FROM HoaDon WHERE NgayBan > ? AND KhachHang.MaKhachHang = HoaDon.MaKhachHang) |
| Thêm mới khách hàng | INSERT INTO KhachHang(MaKhachHang, HoTen, DiemThuong, DienThoai, GioiTinh, NgaySinh, NVQuanLy, Email, DiaChi, GhiChu, TrangThai)  VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?) |
| Cập nhật khách hàng | UPDATE KhachHang SET HoTen = ?, DiemThuong = ?, DienThoai = ?, GioiTinh = ?, NgaySinh = ?, Email = ?, DiaChi = ?, GhiChu = ?, TrangThai = ?  WHERE MaKhachHang = ? |
| Cập nhật trạng thái | UPDATE KhachHang SET TrangThai = ? WHERE MaKhachHang = ? |
| Cập nhật điểm thưởng | UPDATE KhachHang SET DiemThuong = ? WHERE MaKhachHang = ? |

* + - 1. **Bảng NhaCungCap**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn theo mã | SELECT \* FROM NhaCungCap WHERE MaNCC = ? |
| Truy vấn theo email | SELECT \* FROM NhaCungCap WHERE Email LIKE ? |
| Truy vấn theo SĐT | SELECT \* FROM NhaCungCap WHERE DienThoai = ? |
| Truy vấn theo tên hoặc mã hoặc SĐT | SELECT \* FROM NhaCungCap WHERE  MaNCC LIKE ?  OR TenNCC LIKE ?  OR DienThoai LIKE ? |
| Truy vấn theo trạng thái | SELECT \* FROM NhaCungCap WHERE TrangThai = ? |
| Thêm mới nhà cung cấp | INSERT INTO NhaCungCap(MaNCC, TenNCC, DienThoai, Email, DiaChi, TrangThai, GhiChu)  VALUES (?, ?, ?, ?, ?, ?, ?) |
| Cập nhật nhà cung cấp | UPDATE NhaCungCap SET TenNCC = ?,DienThoai = ?, Email = ?, DiaChi = ?, GhiChu = ?, TrangThai = ?  WHERE MaNCC = ? |
| Cập nhật trạng thái | UPDATE NhaCungCap SET TrangThai = ? WHERE MaNCC = ? |

* + - 1. **Bảng NhomHang**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn tất cả | SELECT \* FROM NhomHang |
| Truy vấn theo mã | SELECT \* FROM NhomHang WHERE MaNhomHang = ? |
| Truy vấn theo tên nhóm hàng | SELECT \* FROM NhomHang WHERE TenNhomHang LIKE ? |

* + - 1. **Bảng SanPham**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn tất cả | SELECT \* FROM SanPham |
| Truy vấn theo mã vạch | SELECT \* FROM SanPham WHERE MaVach = ? |
| Truy vấn theo mã | SELECT \* FROM SanPham WHERE MaSanPham = ? |
| Truy vấn theo tên hoặc mã | SELECT \* FROM SanPham WHERE MaSanPham LIKE ? OR TenSanPham LIKE ? |
| Truy vấn theo mã nhà cung cấp | SELECT \* FROM SanPham WHERE MaNCC = ? |
| Truy vấn theo mã nhóm hàng | SELECT \* FROM SanPham WHERE MaNhomHang = ? |
| Thêm mới sản phẩm | INSERT INTO SanPham(MaSanPham, TenSanPham, MaVach, MaNhomHang, MaNCC, MaDVT, TonKho, DinhMucNhoNhat, DinhMucLonNhat, GiaVon, GiaBan, HSDTruoc, HSDMoiNhat, HinhAnh, TrangThai)  VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?) |
| Cập nhật sản phẩm | UPDATE SanPham SET TenSanPham = ?, MaVach = ?, MaNhomHang = ?, MaNCC = ?, MaDVT = ?, DinhMucNhoNhat = ?, DinhMucLonNhat = ?, GiaBan = ?, HSDTruoc = ?, HSDMoiNhat = ?, HinhAnh = ?, TrangThai = ? WHERE MaSanPham = ? |
| Cập nhật tồn kho | UPDATE SanPham SET TonKho = ?  WHERE MaSanPham = ? |
| Cập nhật trạng thái | UPDATE SanPham SET TrangThai = ?  WHERE MaSanPham = ? |

* + - 1. **Bảng HoaDon**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn tất cả | SELECT \* FROM HoaDon |
| Truy vấn theo mã | SELECT \* FROM HoaDon WHERE MaHoaDon = ? |
| Truy vấn theo ngày xuất hoá đơn | SELECT \* FROM HoaDon WHERE NgayBan = ? |
| Thêm mới hoá đơn | {call sp\_insert\_hoadon(?, ?, ?, ?, ?, ?, ?, ?)} |

* + - 1. **Bảng HoaDonChiTiet**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn theo mã | SELECT \* FROM HoaDonChiTiet  WHERE MaHoaDon = ? |
| Thêm mới HĐCT | INSERT INTO HoaDonChiTiet(MaHoaDon, MaSanPham, DonGia, SoLuong, ThanhTien) VALUES (?, ?, ?, ?, ?) |

* + - 1. **Bảng PhieuNhapHang**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn theo mã | SELECT \* FROM PhieuNhapHang  WHERE MaPhieuNhap = ? |
| Truy vấn theo mã NCC | SELECT \* FROM PhieuNhapHang  WHERE MaNCC = ? |
| Truy vấn theo mã sản phẩm hoặc tên sản phẩm | SELECT \* FROM PhieuNhapHang PN WHERE EXISTS  (SELECT MaPhieuNhap FROM PhieuNhapChiTiet PNCT  WHERE PNCT.MaPhieuNhap = PN.MaPhieuNhap  AND EXISTS  (SELECT MaSanPham FROM SanPham SP  WHERE PNCT.MaSanPham = SP.MaSanPham  AND (MaSanPham = ? OR TenSanPham LIKE ?))) |
| Thêm mới phiếu nhập hàng | INSERT INTO PhieuNhapHang(MaPhieuNhap, MaNhanVien, NgayNhap, MaNCC, TongSoLuong, ChietKhau, VAT, ThanhTien, GhiChu) VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?) |
| Cập nhật Ghi chú | UPDATE PhieuNhapHang SET GhiChu = ? WHERE MaPhieuNhap = ? |

* + - 1. **Bảng PhieuNhapChiTiet**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn theo mã phiếu nhập | SELECT \* FROM PhieuNhapChiTiet  WHERE MaPhieuNhap = ? |
| Thêm mới phiếu nhập chi tiết | INSERT INTO PhieuNhapChiTiet(MaPhieuNhap, MaSanPham, SoLuong, GiaNhap, ThanhTien, HSDMoiNhat) VALUES (?, ?, ?, ?, ?, ?) |

* + - 1. **Bảng PhieuKiemKe**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn tất cả | SELECT \* FROM PhieuKiemKe |
| Truy vấn theo mã phiếu kiểm kê | SELECT \* FROM PhieuKiemKe  WHERE MaPhieuKiemKe = ? |
| Truy vấn theo trạng thái | SELECT \* FROM PhieuKiemKe WHERE TrangThai IN ? |
| Truy vấn theo mã sản phẩm hoặc tên sản phầm | SELECT \* FROM PhieuKiemKe PK WHERE EXISTS (SELECT MaPhieuKiem FROM PhieuKiemChiTiet PKCT WHERE PKCT.MaPhieuKiem = PK.MaPhieuKiemKe AND EXISTS (SELECT MaSanPham FROM SanPham SP WHERE PKCT.MaSanPham = SP.MaSanPham AND (MaSanPham = ? OR TenSanPham LIKE ?))) |
| Thêm phiếu kiểm | INSERT INTO PhieuKiemKe(MaPhieuKiemKe, MaNhanVien, NgayTao, GiaTriLech, GhiChu, TrangThai) VALUES (?, ?, ?, ?, ?, ?) |
| Cập nhật phiếu kiểm | {call sp\_update\_phieukiemke(?, ?, ?, ?)} |

* + - 1. **Bảng PhieuKiemChiTiet**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn theo mã phiếu kiểm kê | SELECT \* FROM PhieuKiemChiTiet  WHERE MaPhieuKiem = ? |
| Thêm mới phiếu kiểm chi tiết | INSERT INTO PhieuKiemChiTiet(MaPhieuKiem, MaSanPham, DonGia, DVT, TonKho, ThucTe, SoLuongLech, GiaTriLech) VALUES (?, ?, ?, ?, ?, ?, ?, ?) |
| Cập nhật phiếu kiểm chi tiết | UPDATE PhieuKiemChiTiet SET DonGia = ?, TonKho = ?, ThucTe = ?, SoLuongLech = ?, GiaTriLech = ?  WHERE MaPKCT = ? |

* + - 1. **Bảng PhieuHuyHang**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn tất cả | SELECT \* FROM PhieuHuyHang |
| Truy vấn theo mã phiếu huỷ | SELECT \* FROM PhieuHuyHang  WHERE MaPhieuHuy = ? |
| Truy vấn theo mã sản phẩm hoặc tên sản phẩm | SELECT \* FROM PhieuHuyHang PH WHERE EXISTS  (SELECT MaPhieuHuy FROM PhieuHuyChiTiet PHCT  WHERE PHCT.MaPhieuHuy = PH.MaPhieuHuy  AND EXISTS  (SELECT MaSanPham FROM SanPham SP  WHERE PHCT.MaSanPham = SP.MaSanPham  AND (MaSanPham = ? OR TenSanPham LIKE ?))) |
| Truy vấn theo loại phiếu | SELECT \* FROM PhieuHuyHang  WHERE LoaiPhieuHuy IN ? |
| Thêm mới phiếu huỷ | INSERT INTO PhieuHuyHang(MaPhieuHuy, LoaiPhieuHuy, MaNhanVien, NgayHuy, SoLuongHuy, GiaTriHuy, GhiChu) VALUES (?, ?, ?, ?, ?, ?, ?) |

* + - 1. **Bảng PhieuHuyChiTiet**

|  |  |
| --- | --- |
| **Hành động** | **Câu lệnh SQL** |
| Truy vấn theo mã phiếu huỷ | SELECT \* FROM PhieuHuyChiTiet  WHERE MaPhieuHuy = ? |
| Thêm mới phiếu huỷ chi tiết | INSERT INTO PhieuHuyChiTiet(MaPhieuHuy, MaSanPham, DVT, SoLuong, GiaVon, ThanhTien) VALUES (?, ?, ?, ?, ?, ?) |

### Các thủ tục lưu tổng hợp thống kê

* + - 1. **Tổng kết tình hình cuối ngày bán của nhân viên**

CREATE PROC sp\_tongketcuoingay (

@MaNhanVien nvarchar(20),

@NgayBan nvarchar(10)

)

AS BEGIN

SELECT NH.TenNhomHang AS 'TenNhomHang', SUM(['Bang2'].TongSL) AS 'TongSL', SUM(['Bang2'].TongTT) AS 'TongTT'

FROM

(SELECT ['Bang1'].MaSP AS 'MaSP', SP.MaNhomHang AS 'MaNH', ['Bang1'].TongSL AS 'TongSL', ['Bang1'].TongTT AS 'TongTT'

FROM

(SELECT HDCT.MaSanPham AS 'MaSP', SUM(HDCT.SoLuong) AS 'TongSL', SUM(HDCT.ThanhTien) AS 'TongTT'

FROM HoaDonChiTiet HDCT LEFT JOIN HoaDon HD ON HDCT.MaHoaDon = HD.MaHoaDon

WHERE HD.MaNhanVien = @MaNhanVien AND HD.NgayBan = @NgayBan

GROUP BY HDCT.MaSanPham) AS ['Bang1']

JOIN SanPham SP ON SP.MaSanPham = ['Bang1'].MaSP) AS ['Bang2']

JOIN NhomHang NH ON NH.MaNhomHang = ['Bang2'].MaNH

GROUP BY NH.TenNhomHang

END

* + - 1. **Tính giá vốn hàng bán trong khoảng thời gian**

CREATE PROC [dbo].[sp\_giavonhangban](@tungay date, @denngay date)

AS BEGIN

SELECT SUM([tam])

FROM (SELECT

SUM(SoLuong)\*(SELECT GiaVon FROM SanPham WHERE SanPham.MaSanPham = HoaDonChiTiet.MaSanPham) as [tam]

FROM HoaDonChiTiet

WHERE MaHoaDon IN

(SELECT MaHoaDon FROM HoaDon WHERE NgayBan >= @tungay AND NgayBan <= @denngay)

GROUP BY MaSanPham) AS [bangtam]

END

* + - 1. **Thủ tục thêm mới hoá đơn**

CREATE PROC [dbo].[sp\_insert\_hoadon]

@MaKhachHang nvarchar(7) = null,

@MaNhanVien nvarchar(20) = null,

@NgayBan date = null,

@TongSoLuong int = null,

@ChietKhau decimal(12,3) = null,

@VAT decimal(3,1) = null,

@ThanhTien decimal(12,3) = null,

@TienKhachTra decimal(12,3) = null

AS

BEGIN TRY

BEGIN TRAN

DECLARE @MaHoaDon nvarchar(9)

SET @MaHoaDon = 'HD'+CAST((dbo.fn\_demsohoadon()+1) AS varchar)

INSERT INTO HoaDon(MaHoaDon, MaKhachHang, MaNhanVien, NgayBan, TongSoLuong, ChietKhau, VAT, ThanhTien, TienKhachTra)

VALUES

(@MaHoaDon, @MaKhachHang, @MaNhanVien, @NgayBan, @TongSoLuong, @ChietKhau, @VAT, @ThanhTien, @TienKhachTra)

COMMIT TRAN

END TRY

BEGIN CATCH

ROLLBACK TRAN

END CATCH

* + - 1. **Thống kê doanh thu 7 ngày gần nhất**

CREATE PROC [dbo].[sp\_bc\_doanhthu7ngay]

AS BEGIN

SELECT

(SELECT SUM(ThanhTien) FROM HoaDon WHERE NgayBan = format(getdate()-6,'yyyy-MM-dd')) AS 'ngay6',

(SELECT SUM(ThanhTien) FROM HoaDon WHERE NgayBan = format(getdate()-5,'yyyy-MM-dd')) AS 'ngay5',

(SELECT SUM(ThanhTien) FROM HoaDon WHERE NgayBan = format(getdate()-4,'yyyy-MM-dd')) AS 'ngay4',

(SELECT SUM(ThanhTien) FROM HoaDon WHERE NgayBan = format(getdate()-3,'yyyy-MM-dd')) AS 'ngay3',

(SELECT SUM(ThanhTien) FROM HoaDon WHERE NgayBan = format(getdate()-2,'yyyy-MM-dd')) AS 'ngay2',

(SELECT SUM(ThanhTien) FROM HoaDon WHERE NgayBan = format(getdate()-1,'yyyy-MM-dd')) AS 'ngay1',

(SELECT SUM(ThanhTien) FROM HoaDon WHERE NgayBan = format(getdate(),'yyyy-MM-dd')) AS 'ngay0',

END

* + - 1. **Thống kê hàng bán chạy theo khoảng ngày**

CREATE PROC [dbo].[sp\_tk\_hangbanchay](@tungay date, @denngay date)

AS BEGIN

SELECT TOP(10) HDCT.MaSanPham,

(SELECT TenSanPham FROM SanPham WHERE SanPham.MaSanPham = HDCT.MaSanPham) AS 'tensp',

(SELECT MoTa FROM (SELECT MaSanPham, MoTa

FROM SanPham JOIN DonViTinh ON SanPham.MaDVT = DonViTinh.MaDVT

WHERE EXISTS (SELECT MaSanPham FROM HoaDonChiTiet WHERE HoaDonChiTiet.MaSanPham = SanPham.MaSanPham)) AS [bang1]

WHERE HDCT.MaSanPham = [bang1].MaSanPham),

COUNT(HDCT.MaSanPham)

FROM HoaDonChiTiet HDCT

WHERE HDCT.MaHoaDon IN (SELECT MaHoaDon FROM HoaDonChiTiet

WHERE EXISTS

(SELECT \* FROM HoaDon

WHERE NgayBan >= @tungay

AND NgayBan <= @denngay

AND HoaDonChiTiet.MaHoaDon = HoaDon.MaHoaDon))

GROUP BY HDCT.MaSanPham

ORDER BY COUNT(HDCT.MaSanPham) DESC

END

* + - 1. **Thống kê hàng bán chạy toàn thời gian**

CREATE PROC [dbo].[sp\_tk\_hangbanchay\_toantg]

AS BEGIN

select top(10)

hdct.masanpham,

(select tensanpham from sanpham where sanpham.masanpham = hdct.masanpham) as 'tensp',

(select mota from (select masanpham, mota

from sanpham join donvitinh on sanpham.madvt = donvitinh.madvt

where exists

(select masanpham from hoadonchitiet where hoadonchitiet.masanpham = sanpham.masanpham))

as [bang1] where hdct.masanpham = [bang1].masanpham),

count(hdct.masanpham)

from hoadonchitiet hdct

group by hdct.masanpham

order by count(hdct.masanpham) desc

END

* + - 1. **Thống kê tổng hợp hàng nhập**

CREATE PROC [dbo].[sp\_tk\_tonghophangnhap]

AS BEGIN

select pnct.masanpham,

(select tensanpham from sanpham where sanpham.masanpham = pnct.masanpham) as 'tensp',

(select mota from donvitinh where madvt in (select madvt from sanpham where sanpham.masanpham = pnct.masanpham)) as 'dvt',

sum(pnct.soluong),

sum(pnct.thanhtien)

from phieunhapchitiet pnct

group by pnct.masanpham

ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)

END

* + - 1. **Thống kê hàng nhập theo khoảng thời gian**

CREATE PROC [dbo].[sp\_tk\_tonghophangnhap\_theotg](@tungay date, @denngay date)

AS BEGIN

select pnct.masanpham,

(select tensanpham from sanpham where sanpham.masanpham = pnct.masanpham) as 'tensp',

(select mota from donvitinh where madvt in (select madvt from sanpham where sanpham.masanpham = pnct.masanpham)) as 'dvt',

sum(pnct.soluong),

sum(pnct.thanhtien)

from phieunhapchitiet pnct

where pnct.maphieunhap in (select maphieunhap from phieunhaphang where ngaynhap >= @tungay and ngaynhap <= @denngay)

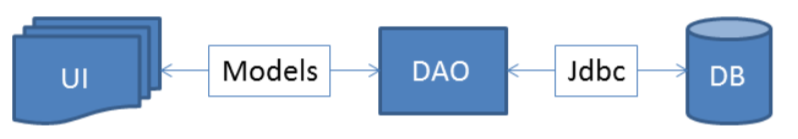
group by pnct.masanpham

ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)

END

## LẬP TRÌNH JDBC

Trong dự án này chúng ta chọn mô hình lập trình với CSDL như hình sau



Trong đó:

* UI: các thành phần giao diện
* DB: cơ sở dữ liệu
* Jdbc: là lớp tiện ích cung cấp các hàm tiện ích làm việc với CSDL thông qua câu lệnh SQL hoặc lời gọi thủ tục lưu
* Model: là các lớp mô tả dữ liệu theo cấu trúc các bảng trong CSDL
* DAO: là các lớp thao tác và truy vấn dữ liệu. Nó có nhiệm vụ chuyển đổi Model sang SQL và ngược lại.

Theo mô hình này thì lập trình giao diện làm việc với CSDL thông qua các lớp model. Model như là một gói dữ liệu được sử dụng để giao tiếp giữa giao diện và các thành phần DAO. DAO mới thực sự là lập trình CSDL nó làm nhiệm vụ nhận model từ giao diện, chuyển đổi sang SQL và sử dụng Jdbc để thực hiện các thao tác dữ liệu. Ngược lại thì truy vấn dữ liệu từ CSDL, chuyển đổi thành model để hiển thị lên các thành phần giao diện.

Việc tách thành các phần riêng biết giúp quản lý dự án tốt hơn, dễ bảo trì nâng cấp hơn.

### Lớp hỗ trợ

* + - 1. **Lớp tiện ích jdbcHelper**

package helper;

import java.io.File;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.HashMap;

import net.sf.jasperreports.engine.JasperCompileManager;

import net.sf.jasperreports.engine.JasperFillManager;

import net.sf.jasperreports.engine.JasperPrint;

import net.sf.jasperreports.engine.JasperPrintManager;

import net.sf.jasperreports.engine.JasperReport;

import net.sf.jasperreports.engine.design.JasperDesign;

import net.sf.jasperreports.engine.xml.JRXmlLoader;

/\*\*

\*

\* @author Administrator

\*/

public class jdbcHelper {

private static String driver = "com.microsoft.sqlserver.jdbc.SQLServerDriver";

private static String db\_url = "jdbc:sqlserver://localhost:1433;databaseName=quanlysieuthimini";

private static String db\_user = "sa";

private static String db\_pass = "songlong";

/\*

\* Nap driver

\*/

static {

try {

Class.forName(driver);

} catch (ClassNotFoundException e) {

throw new RuntimeException (e);

}

}

/\*\*

\* Xay dung PreparedStatement

\*/

public static PreparedStatement prepareStatement(String sql, Object...args) throws SQLException{

Connection con = DriverManager.getConnection(db\_url, db\_user, db\_pass);

PreparedStatement pstm = null;

if(sql.trim().startsWith("|")){

pstm = con.prepareCall(sql);

}

else{

pstm = con.prepareStatement(sql);

}

for (int i = 0; i < args.length; i++) {

pstm.setObject(i + 1, args[i]);

}

return pstm;

}

/\*\*

\* Thuc hien cac cau lenh SQL thao tac (INSERT, UPDATE, DELETE)

\* hoac thu tuc luu thao tac du lieu

\* @return int - ket qua tra ve so dong da tac dong tren CSDL

\* neu tra ve 0 tuc la khong co thay doi CSDL

\* neu tra ve > 0 tuc la da co thay doi CSDL

\*/

public static int executeUpdate(String sql, Object... args) {

int rs = 0;

try {

PreparedStatement pstm = prepareStatement(sql, args);

try {

rs = pstm.executeUpdate();

return rs;

} finally {

pstm.getConnection().close();

return rs;

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

}

/\*\*

\* Thuc hien cau lenh SQL truy van SELECT

\* hoac thu tuc luu truy van du lieu

\*/

public static ResultSet executeQuery(String sql, Object...args){

try {

PreparedStatement pstm = prepareStatement(sql, args);

return pstm.executeQuery();

} catch (SQLException e) {

throw new RuntimeException (e);

}

}

public static ResultSet excuteQuery(String sql, Object[] args) {

try {

PreparedStatement pstm = prepareStatement(sql, args);

return pstm.executeQuery();

} catch (SQLException e) {

throw new RuntimeException (e);

}

}

/\*

\* Hàm kết nối ireport và in hoá đơn

\*/

public static void inHoaDon(HashMap HoaDon) throws Exception {

Connection con = DriverManager.getConnection(db\_url, db\_user, db\_pass);

File path = new File(System.getProperty("user.dir")+"\\src\\ui\\chung\\InHoaDon.jrxml");

JasperDesign jdesign = JRXmlLoader.load(path.getAbsolutePath());

// JasperDesign jdesign = JRXmlLoader.load("D:\\FPOLY\\FALL 19\\Du an 1\\quanlysieuthimini\\quanlysieuthimini\\src\\ui\\chung\\InHoaDon.jrxml");

JasperReport ireport = JasperCompileManager.compileReport(jdesign);

JasperPrint jprint = JasperFillManager.fillReport(ireport, HoaDon, con);

// hàm view:

// JasperViewer.viewReport(jprint);

// hàm in:

JasperPrintManager.printReport(jprint, false);

}

}

* + - 1. **Lớp tiện ích DateHelper**

package helper;

import java.text.ParseException;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.concurrent.TimeUnit;

/\*\*

\*

\* @author Administrator

\*/

public class DateHelper {

static final SimpleDateFormat DATE\_FORMATER = new SimpleDateFormat("MM/dd/yyyy");

/\*\*

\* Chuyển đổi String sang Date

\*

\* @param date là String cần chuyển

\* @param pattern là định dạng thời gian

\* @return Date kết quả

\*/

public static Date toDate(String date, String... pattern) {

try {

if (pattern.length > 0) {

DATE\_FORMATER.applyPattern(pattern[0]);

}

if (date == null) {

return DateHelper.now();

}

return DATE\_FORMATER.parse(date);

//return DATE\_FORMATER.parse(date);

} catch (ParseException ex) {

throw new RuntimeException(ex);

}

}

/\*\*

\* Chuyển đổi từ Date sang String

\*

\* @param date là Date cần chuyển đổi

\* @param pattern là định dạng thời gian

\* @return String kết quả

\*/

public static String toString(Date date, String... pattern) {

if (pattern.length > 0) {

DATE\_FORMATER.applyPattern(pattern[0]);

}

if (date == null) {

date = DateHelper.now();

}

return DATE\_FORMATER.format(date);

}

/\*\*

\* Lấy thời gian hiện tại

\*

\* @return Date kết quả

\*/

public static Date now() {

return new Date();

}

/\*\*

\* Bổ sung số ngày vào thời gian

\*

\* @param date thời gian hiện có

\* @param days số ngày cần bổ sung vào date

\* @return Date kết quả

\*/

public static Date addDays(Date date, int days) {

date.setTime(date.getTime() + days \* 24 \* 60 \* 60 \* 1000);

return date;

}

/\*\*

\* Bổ sung số ngày vào thời gian hiện hành

\*

\* @param days số ngày cần bổ sung vào thời gian hiện tại

\* @return Date kết quả

\*/

public static Date add(int days) {

Date now = DateHelper.now();

now.setTime(now.getTime() + days \* 24 \* 60 \* 60 \* 1000);

return now;

}

/\*\*

\* Hàm thực hiện phép tính trừ hai ngày

\*

\* @param date1 - ngày trừ

\* @param date2 - ngày bị trừ

\* @param timeUnit - kiểu kết quả trả về

\* @return số năm

\*/

public static long getDateDiff(Date date1, Date date2, TimeUnit timeUnit) {

long diffInMillies = date2.getTime() - date1.getTime();

return timeUnit.convert(diffInMillies, TimeUnit.MILLISECONDS);

}

/\*

\* Hàm tính thứ trong tuần và trả về ngày của thứ hai

- i = 0: dd-MM-yyyy

- i = 1: yyyy-MM-dd

\*/

public static String tinhNgayDauTuan(int i) {

String getThu = new SimpleDateFormat("EE").format(DateHelper.now());

Date date = DateHelper.now();

switch (getThu) {

case "Mon":

break;

case "Tue":

date = DateHelper.addDays(DateHelper.now(), -1);

break;

case "Wed":

date = DateHelper.addDays(DateHelper.now(), -2);

break;

case "Thu":

date = DateHelper.addDays(DateHelper.now(), -3);

break;

case "Fri":

date = DateHelper.addDays(DateHelper.now(), -4);

break;

case "Sat":

date = DateHelper.addDays(DateHelper.now(), -5);

break;

case "Sun":

date = DateHelper.addDays(DateHelper.now(), -6);

break;

}

if (i == 0)

return DateHelper.toString(date, "dd-MM-yyyy");

else

return DateHelper.toString(date, "yyyy-MM-dd");

}

/\*

\* Hàm tính quý thứ mấy trong năm

quý 1: tháng 1 đến tháng 3

quý 2: tháng 4 đến tháng 6

quý 3: tháng 7 đến tháng 9

quý 4: tháng 10 đến tháng 12

\*/

public static int tinhSoQuyTrongNam() {

int Quy = 0;

String NgayHienTai = DateHelper.toString(DateHelper.now(), "dd-MM-yyyy");

String ThangHienTai = NgayHienTai.substring(3, 5);

switch (ThangHienTai) {

case "01": case "02": case "03":

Quy = 1;

break;

case "04": case "05": case "06":

Quy = 2;

break;

case "07": case "08": case "09":

Quy = 3;

break;

case "10": case "11": case "12":

Quy = 4;

break;

}

return Quy;

}

/\*

\* Hàm kiểm tra năm nhuận

true: nhuận

false: không nhuận

\*/

public static boolean xetNamNhuan(int Nam) {

if (Nam % 4 == 0) {

if (Nam % 100 == 0) {

return Nam % 400 == 0;

} else {

return true;

}

} else {

return false;

}

}

}

### Model class – Các lớp mô tả dữ liệu

* + - 1. **DonViTinh**

package model;

/\*\*

\*

\* @author Administrator

\*/

public class DonViTinh {

private int MaDVT;

private String MoTa;

public DonViTinh() {

}

public DonViTinh(int MaDVT, String MoTa) {

this.MaDVT = MaDVT;

this.MoTa = MoTa;

}

public int getMaDVT() {

return MaDVT;

}

public void setMaDVT(int MaDVT) {

this.MaDVT = MaDVT;

}

public String getMoTa() {

return MoTa;

}

public void setMoTa(String MoTa) {

this.MoTa = MoTa;

}

@Override

public String toString() {

return this.MoTa;

}

}

* + - 1. **NhomHang**

package model;

/\*\*

\*

\* @author Administrator

\*/

public class NhomHang {

private String MaNhomHang;

private String TenNhomHang;

private String GhiChu;

public NhomHang(){

}

public NhomHang(String MaNhomHang, String TenNhomHang, String GhiChu) {

this.MaNhomHang = MaNhomHang;

this.TenNhomHang = TenNhomHang;

this.GhiChu = GhiChu;

}

public String getMaNhomHang() {

return MaNhomHang;

}

public void setMaNhomHang(String MaNhomHang) {

this.MaNhomHang = MaNhomHang;

}

public String getTenNhomHang() {

return TenNhomHang;

}

public void setTenNhomHang(String TenNhomHang) {

this.TenNhomHang = TenNhomHang;

}

public String getGhiChu() {

return GhiChu;

}

public void setGhiChu(String GhiChu) {

this.GhiChu = GhiChu;

}

@Override

public String toString() {

return this.TenNhomHang;

}

}

* + - 1. **NhaCungCap**

package model;

/\*\*

\*

\* @author Administrator

\*/

public class NhaCungCap {

private String MaNCC;

private String TenNCC;

private String DienThoai;

private String Email;

private String DiaChi;

private String GhiChu;

private boolean TrangThai;

public NhaCungCap() {

}

public NhaCungCap(String MaNCC, String TenNCC, String DienThoai, String Email, String DiaChi, String GhiChu, boolean TrangThai) {

this.MaNCC = MaNCC;

this.TenNCC = TenNCC;

this.DienThoai = DienThoai;

this.Email = Email;

this.DiaChi = DiaChi;

this.GhiChu = GhiChu;

this.TrangThai = TrangThai;

}

public String getMaNCC() {

return MaNCC;

}

public void setMaNCC(String MaNCC) {

this.MaNCC = MaNCC;

}

public String getTenNCC() {

return TenNCC;

}

public void setTenNCC(String TenNCC) {

this.TenNCC = TenNCC;

}

public String getDienThoai() {

return DienThoai;

}

public void setDienThoai(String DienThoai) {

this.DienThoai = DienThoai;

}

public String getEmail() {

return Email;

}

public void setEmail(String Email) {

this.Email = Email;

}

public String getDiaChi() {

return DiaChi;

}

public void setDiaChi(String DiaChi) {

this.DiaChi = DiaChi;

}

public String getGhiChu() {

return GhiChu;

}

public void setGhiChu(String GhiChu) {

this.GhiChu = GhiChu;

}

public boolean getTrangThai() {

return TrangThai;

}

public void setTrangThai(boolean TrangThai) {

this.TrangThai = TrangThai;

}

@Override

public String toString() {

return this.TenNCC;

}

}

* + - 1. **NguoiDung**

package model;

import java.util.Date;

/\*\*

\*

\* @author Administrator

\*/

public class NguoiDung {

private String TenDangNhap;

private String MatKhau;

private String HoTen;

private String DienThoai;

private boolean GioiTinh;

private Date NgaySinh;

private String Email;

private String DiaChi;

private int VaiTro;

private boolean TrangThai;

public NguoiDung() {

}

public NguoiDung(String TenDangNhap, String MatKhau, String HoTen, String DienThoai, boolean GioiTinh, Date NgaySinh, String Email, String DiaChi, int VaiTro, boolean TrangThai) {

this.TenDangNhap = TenDangNhap;

this.MatKhau = MatKhau;

this.HoTen = HoTen;

this.DienThoai = DienThoai;

this.GioiTinh = GioiTinh;

this.NgaySinh = NgaySinh;

this.Email = Email;

this.DiaChi = DiaChi;

this.VaiTro = VaiTro;

this.TrangThai = TrangThai;

}

public String getTenDangNhap() {

return TenDangNhap;

}

public void setTenDangNhap(String TenDangNhap) {

this.TenDangNhap = TenDangNhap;

}

public String getMatKhau() {

return MatKhau;

}

public void setMatKhau(String MatKhau) {

this.MatKhau = MatKhau;

}

public String getHoTen() {

return HoTen;

}

public void setHoTen(String HoTen) {

this.HoTen = HoTen;

}

public String getDienThoai() {

return DienThoai;

}

public void setDienThoai(String DienThoai) {

this.DienThoai = DienThoai;

}

public boolean getGioiTinh() {

return GioiTinh;

}

public void setGioiTinh(boolean GioiTinh) {

this.GioiTinh = GioiTinh;

}

public Date getNgaySinh() {

return NgaySinh;

}

public void setNgaySinh(Date NgaySinh) {

this.NgaySinh = NgaySinh;

}

public String getEmail() {

return Email;

}

public void setEmail(String Email) {

this.Email = Email;

}

public String getDiaChi() {

return DiaChi;

}

public void setDiaChi(String DiaChi) {

this.DiaChi = DiaChi;

}

public int getVaiTro() {

return VaiTro;

}

public void setVaiTro(int VaiTro) {

this.VaiTro = VaiTro;

}

public boolean getTrangThai() {

return TrangThai;

}

public void setTrangThai(boolean TrangThai) {

this.TrangThai = TrangThai;

}

}

* + - 1. **KhachHang**

package model;

import java.util.Date;

/\*\*

\*

\* @author Administrator

\*/

public class KhachHang {

private String MaKhachHang;

private String HoTen;

private int DiemThuong;

private String DienThoai;

private boolean GioiTinh;

private Date NgaySinh;

private String NVQuanLy;

private String Email;

private String DiaChi;

private String GhiChu;

private boolean TrangThai;

public KhachHang(){

}

public KhachHang(String MaKhachHang, String HoTen, int DiemThuong, String DienThoai, boolean GioiTinh, Date NgaySinh, String NVQuanLy, String Email, String DiaChi, String GhiChu, boolean TrangThai) {

this.MaKhachHang = MaKhachHang;

this.HoTen = HoTen;

this.DiemThuong = DiemThuong;

this.DienThoai = DienThoai;

this.GioiTinh = GioiTinh;

this.NgaySinh = NgaySinh;

this.NVQuanLy = NVQuanLy;

this.Email = Email;

this.DiaChi = DiaChi;

this.GhiChu = GhiChu;

this.TrangThai = TrangThai;

}

public String getMaKhachHang() {

return MaKhachHang;

}

public void setMaKhachHang(String MaKhachHang) {

this.MaKhachHang = MaKhachHang;

}

public String getHoTen() {

return HoTen;

}

public void setHoTen(String HoTen) {

this.HoTen = HoTen;

}

public int getDiemThuong() {

return DiemThuong;

}

public void setDiemThuong(int DiemThuong) {

this.DiemThuong = DiemThuong;

}

public String getDienThoai() {

return DienThoai;

}

public void setDienThoai(String DienThoai) {

this.DienThoai = DienThoai;

}

public boolean getGioiTinh() {

return GioiTinh;

}

public void setGioiTinh(boolean GioiTinh) {

this.GioiTinh = GioiTinh;

}

public Date getNgaySinh() {

return NgaySinh;

}

public void setNgaySinh(Date NgaySinh) {

this.NgaySinh = NgaySinh;

}

public String getNVQuanLy() {

return NVQuanLy;

}

public void setNVQuanLy(String NVQuanLy) {

this.NVQuanLy = NVQuanLy;

}

public String getEmail() {

return Email;

}

public void setEmail(String Email) {

this.Email = Email;

}

public String getDiaChi() {

return DiaChi;

}

public void setDiaChi(String DiaChi) {

this.DiaChi = DiaChi;

}

public String getGhiChu() {

return GhiChu;

}

public void setGhiChu(String GhiChu) {

this.GhiChu = GhiChu;

}

public boolean getTrangThai() {

return TrangThai;

}

public void setTrangThai(boolean TrangThai) {

this.TrangThai = TrangThai;

}

}

* + - 1. **SanPham**

package model;

import java.util.Date;

/\*\*

\*

\* @author Administrator

\*/

public class SanPham implements Comparable<SanPham>{

private String MaSanPham;

private String TenSanPham;

private String MaVach;

private String MaNhomHang;

private String MaNCC;

private int MaDVT;

private int TonKho;

private int DinhMucNhoNhat;

private int DinhMucLonNhat;

private float GiaVon;

private float GiaBan;

private Date HSDTruoc;

private Date HSDMoiNhat;

private String HinhAnh;

private boolean TrangThai;

public SanPham(){

}

public SanPham(String MaSanPham, String TenSanPham, String MaVach, String MaNhomHang, String MaNCC, int MaDVT, int TonKho, int DinhMucNhoNhat, int DinhMucLonNhat, float GiaVon, float GiaBan, Date HSDTruoc, Date HSDMoiNhat, String HinhAnh, boolean TrangThai) {

this.MaSanPham = MaSanPham;

this.TenSanPham = TenSanPham;

this.MaVach = MaVach;

this.MaNhomHang = MaNhomHang;

this.MaNCC = MaNCC;

this.MaDVT = MaDVT;

this.TonKho = TonKho;

this.DinhMucNhoNhat = DinhMucNhoNhat;

this.DinhMucLonNhat = DinhMucLonNhat;

this.GiaVon = GiaVon;

this.GiaBan = GiaBan;

this.HSDTruoc = HSDTruoc;

this.HSDMoiNhat = HSDMoiNhat;

this.HinhAnh = HinhAnh;

this.TrangThai = TrangThai;

}

public String getMaSanPham() {

return MaSanPham;

}

public void setMaSanPham(String MaSanPham) {

this.MaSanPham = MaSanPham;

}

public String getTenSanPham() {

return TenSanPham;

}

public void setTenSanPham(String TenSanPham) {

this.TenSanPham = TenSanPham;

}

public String getMaVach() {

return MaVach;

}

public void setMaVach(String MaVach) {

this.MaVach = MaVach;

}

public String getMaNhomHang() {

return MaNhomHang;

}

public void setMaNhomHang(String MaNhomHang) {

this.MaNhomHang = MaNhomHang;

}

public String getMaNCC() {

return MaNCC;

}

public void setMaNCC(String MaNCC) {

this.MaNCC = MaNCC;

}

public int getMaDVT() {

return MaDVT;

}

public void setMaDVT(int MaDVT) {

this.MaDVT = MaDVT;

}

public int getTonKho() {

return TonKho;

}

public void setTonKho(int TonKho) {

this.TonKho = TonKho;

}

public int getDinhMucNhoNhat() {

return DinhMucNhoNhat;

}

public void setDinhMucNhoNhat(int DinhMucNhoNhat) {

this.DinhMucNhoNhat = DinhMucNhoNhat;

}

public int getDinhMucLonNhat() {

return DinhMucLonNhat;

}

public void setDinhMucLonNhat(int DinhMucLonNhat) {

this.DinhMucLonNhat = DinhMucLonNhat;

}

public float getGiaVon() {

return GiaVon;

}

public void setGiaVon(float GiaVon) {

this.GiaVon = GiaVon;

}

public float getGiaBan() {

return GiaBan;

}

public void setGiaBan(float GiaBan) {

this.GiaBan = GiaBan;

}

public Date getHSDTruoc() {

return HSDTruoc;

}

public void setHSDTruoc(Date HSDTruoc) {

this.HSDTruoc = HSDTruoc;

}

public Date getHSDMoiNhat() {

return HSDMoiNhat;

}

public void setHSDMoiNhat(Date HSDMoiNhat) {

this.HSDMoiNhat = HSDMoiNhat;

}

public String getHinhAnh() {

return HinhAnh;

}

public void setHinhAnh(String HinhAnh) {

this.HinhAnh = HinhAnh;

}

public boolean getTrangThai() {

return TrangThai;

}

public void setTrangThai(boolean TrangThai) {

this.TrangThai = TrangThai;

}

@Override

public int compareTo(SanPham o) {

return this.MaSanPham.compareTo(o.MaSanPham);

}

}

* + - 1. **PhieuNhap**

package model;

import java.util.Date;

import java.util.List;

import java.util.TreeMap;

/\*\*

\*

\* @author Administrator

\*/

public class PhieuNhap {

private String MaPhieuNhap;

private String MaNhanVien;

private Date NgayNhap;

private String MaNCC;

private int TongSoLuong;

private float ChietKhau;

private float VAT;

private double ThanhTien;

private String GhiChu;

private TreeMap<SanPham, List<String>> DSSP = new TreeMap<>();

public PhieuNhap() {

}

public PhieuNhap(String MaPhieuNhap, String MaNhanVien, Date NgayNhap, String MaNCC, int TongSoLuong, float ChietKhau, float VAT, double ThanhTien, String GhiChu) {

this.MaPhieuNhap = MaPhieuNhap;

this.MaNhanVien = MaNhanVien;

this.NgayNhap = NgayNhap;

this.MaNCC = MaNCC;

this.TongSoLuong = TongSoLuong;

this.ChietKhau = ChietKhau;

this.VAT = VAT;

this.ThanhTien = ThanhTien;

this.GhiChu = GhiChu;

}

public String getMaPhieuNhap() {

return MaPhieuNhap;

}

public void setMaPhieuNhap(String MaPhieuNhap) {

this.MaPhieuNhap = MaPhieuNhap;

}

public String getMaNhanVien() {

return MaNhanVien;

}

public void setMaNhanVien(String MaNhanVien) {

this.MaNhanVien = MaNhanVien;

}

public Date getNgayNhap() {

return NgayNhap;

}

public void setNgayNhap(Date NgayNhap) {

this.NgayNhap = NgayNhap;

}

public String getMaNCC() {

return MaNCC;

}

public void setMaNCC(String MaNCC) {

this.MaNCC = MaNCC;

}

public int getTongSoLuong() {

return TongSoLuong;

}

public void setTongSoLuong(int TongSoLuong) {

this.TongSoLuong = TongSoLuong;

}

public float getChietKhau() {

return ChietKhau;

}

public void setChietKhau(float ChietKhau) {

this.ChietKhau = ChietKhau;

}

public float getVAT() {

return VAT;

}

public void setVAT(float VAT) {

this.VAT = VAT;

}

public double getThanhTien() {

return ThanhTien;

}

public void setThanhTien(double ThanhTien) {

this.ThanhTien = ThanhTien;

}

public String getGhiChu() {

return GhiChu;

}

public void setGhiChu(String GhiChu) {

this.GhiChu = GhiChu;

}

public TreeMap<SanPham, List<String>> getDSSP() {

return DSSP;

}

public void setDSSP(TreeMap<SanPham, List<String>> DSSP) {

this.DSSP = DSSP;

}

}

* + - 1. **PhieuNhapChiTiet**

package model;

import java.util.Date;

/\*\*

\*

\* @author Administrator

\*/

public class PhieuNhapChiTiet {

private int MaPNCT;

private String MaPhieuNhap;

private String MaSanPham;

private int SoLuong;

private float GiaNhap;

private double ThanhTien;

private Date HSDMoiNhat;

public PhieuNhapChiTiet() {

}

public PhieuNhapChiTiet(int MaPNCT, String MaPhieuNhap, String MaSanPham, int SoLuong, float GiaNhap, double ThanhTien, Date HSDMoiNhat) {

this.MaPNCT = MaPNCT;

this.MaPhieuNhap = MaPhieuNhap;

this.MaSanPham = MaSanPham;

this.SoLuong = SoLuong;

this.GiaNhap = GiaNhap;

this.ThanhTien = ThanhTien;

this.HSDMoiNhat = HSDMoiNhat;

}

public int getMaPNCT() {

return MaPNCT;

}

public void setMaPNCT(int MaPNCT) {

this.MaPNCT = MaPNCT;

}

public String getMaPhieuNhap() {

return MaPhieuNhap;

}

public void setMaPhieuNhap(String MaPhieuNhap) {

this.MaPhieuNhap = MaPhieuNhap;

}

public String getMaSanPham() {

return MaSanPham;

}

public void setMaSanPham(String MaSanPham) {

this.MaSanPham = MaSanPham;

}

public int getSoLuong() {

return SoLuong;

}

public void setSoLuong(int SoLuong) {

this.SoLuong = SoLuong;

}

public float getGiaNhap() {

return GiaNhap;

}

public void setGiaNhap(float GiaNhap) {

this.GiaNhap = GiaNhap;

}

public double getThanhTien() {

return ThanhTien;

}

public void setThanhTien(double ThanhTien) {

this.ThanhTien = ThanhTien;

}

public Date getHSDMoiNhat() {

return HSDMoiNhat;

}

public void setHSDMoiNhat(Date HSDMoiNhat) {

this.HSDMoiNhat = HSDMoiNhat;

}

}

* + - 1. **PhieuHuyHang**

package model;

import java.util.Date;

import java.util.TreeMap;

/\*\*

\*

\* @author Administrator

\*/

public class PhieuHuyHang {

private String MaPhieuHuy;

private int LoaiPhieuHuy; // 0: huỷ hết hsd; 1: huỷ lỗi

private String MaNhanVien;

private Date NgayHuy;

private int SoLuongHuy;

private float GiaTriHuy;

private String GhiChu;

private TreeMap<SanPham, Integer> DSSP;

public PhieuHuyHang() {

}

public PhieuHuyHang(String MaPhieuHuy, int LoaiPhieuHuy, String MaNhanVien, Date NgayHuy, int SoLuongHuy, float GiaTriHuy, String GhiChu, TreeMap<SanPham, Integer> DSSP) {

this.MaPhieuHuy = MaPhieuHuy;

this.LoaiPhieuHuy = LoaiPhieuHuy;

this.MaNhanVien = MaNhanVien;

this.NgayHuy = NgayHuy;

this.SoLuongHuy = SoLuongHuy;

this.GiaTriHuy = GiaTriHuy;

this.GhiChu = GhiChu;

this.DSSP = DSSP;

}

public String getMaPhieuHuy() {

return MaPhieuHuy;

}

public void setMaPhieuHuy(String MaPhieuHuy) {

this.MaPhieuHuy = MaPhieuHuy;

}

public int getLoaiPhieuHuy() {

return LoaiPhieuHuy;

}

public void setLoaiPhieuHuy(int LoaiPhieuHuy) {

this.LoaiPhieuHuy = LoaiPhieuHuy;

}

public String getMaNhanVien() {

return MaNhanVien;

}

public void setMaNhanVien(String MaNhanVien) {

this.MaNhanVien = MaNhanVien;

}

public Date getNgayHuy() {

return NgayHuy;

}

public void setNgayHuy(Date NgayHuy) {

this.NgayHuy = NgayHuy;

}

public int getSoLuongHuy() {

return SoLuongHuy;

}

public void setSoLuongHuy(int SoLuongHuy) {

this.SoLuongHuy = SoLuongHuy;

}

public float getGiaTriHuy() {

return GiaTriHuy;

}

public void setGiaTriHuy(float GiaTriHuy) {

this.GiaTriHuy = GiaTriHuy;

}

public String getGhiChu() {

return GhiChu;

}

public void setGhiChu(String GhiChu) {

this.GhiChu = GhiChu;

}

public TreeMap<SanPham, Integer> getDSSP() {

return DSSP;

}

public void setDSSP(TreeMap<SanPham, Integer> DSSP) {

this.DSSP = DSSP;

}

}

* + - 1. **PhieuHuyChiTiet**

package model;

/\*\*

\*

\* @author Administrator

\*/

public class PhieuHuyChiTiet {

private int MaPHCT;

private String MaPhieuHuy;

private String MaSanPham;

private String DVT;

private int SoLuong;

private float GiaVon;

private float ThanhTien;

public PhieuHuyChiTiet() {

}

public PhieuHuyChiTiet(int MaPHCT, String MaPhieuHuy, String MaSanPham, String DVT, int SoLuong, float GiaVon, float ThanhTien) {

this.MaPHCT = MaPHCT;

this.MaPhieuHuy = MaPhieuHuy;

this.MaSanPham = MaSanPham;

this.DVT = DVT;

this.SoLuong = SoLuong;

this.GiaVon = GiaVon;

this.ThanhTien = ThanhTien;

}

public int getMaPHCT() {

return MaPHCT;

}

public void setMaPHCT(int MaPHCT) {

this.MaPHCT = MaPHCT;

}

public String getMaPhieuHuy() {

return MaPhieuHuy;

}

public void setMaPhieuHuy(String MaPhieuHuy) {

this.MaPhieuHuy = MaPhieuHuy;

}

public String getMaSanPham() {

return MaSanPham;

}

public void setMaSanPham(String MaSanPham) {

this.MaSanPham = MaSanPham;

}

public String getDVT() {

return DVT;

}

public void setDVT(String DVT) {

this.DVT = DVT;

}

public int getSoLuong() {

return SoLuong;

}

public void setSoLuong(int SoLuong) {

this.SoLuong = SoLuong;

}

public float getGiaVon() {

return GiaVon;

}

public void setGiaVon(float GiaVon) {

this.GiaVon = GiaVon;

}

public float getThanhTien() {

return ThanhTien;

}

public void setThanhTien(float ThanhTien) {

this.ThanhTien = ThanhTien;

}

}

* + - 1. **PhieuKiem**

package model;

import java.util.Date;

import java.util.TreeMap;

/\*\*

\*

\* @author Administrator

\*/

public class PhieuKiem {

private String MaPhieuKiem;

private String MaNhanVien;

private Date NgayTao;

private Date NgayCanBang;

private double GiaTriLech;

private String GhiChu;

private int TrangThai; // 0: chua can bang; 1: da can bang; 2: da huy

private TreeMap<SanPham,Integer> DSSP;

public PhieuKiem() {

}

public PhieuKiem(String MaPhieuKiem, String MaNhanVien, Date NgayTao, Date NgayCanBang, double GiaTriLech, String GhiChu, int TrangThai, TreeMap<SanPham, Integer> DSSP) {

this.MaPhieuKiem = MaPhieuKiem;

this.MaNhanVien = MaNhanVien;

this.NgayTao = NgayTao;

this.NgayCanBang = NgayCanBang;

this.GiaTriLech = GiaTriLech;

this.GhiChu = GhiChu;

this.TrangThai = TrangThai;

this.DSSP = DSSP;

}

public String getMaPhieuKiem() {

return MaPhieuKiem;

}

public void setMaPhieuKiem(String MaPhieuKiem) {

this.MaPhieuKiem = MaPhieuKiem;

}

public String getMaNhanVien() {

return MaNhanVien;

}

public void setMaNhanVien(String MaNhanVien) {

this.MaNhanVien = MaNhanVien;

}

public Date getNgayTao() {

return NgayTao;

}

public void setNgayTao(Date NgayTao) {

this.NgayTao = NgayTao;

}

public Date getNgayCanBang() {

return NgayCanBang;

}

public void setNgayCanBang(Date NgayCanBang) {

this.NgayCanBang = NgayCanBang;

}

public double getGiaTriLech() {

return GiaTriLech;

}

public void setGiaTriLech(double GiaTriLech) {

this.GiaTriLech = GiaTriLech;

}

public String getGhiChu() {

return GhiChu;

}

public void setGhiChu(String GhiChu) {

this.GhiChu = GhiChu;

}

public int getTrangThai() {

return TrangThai;

}

public void setTrangThai(int TrangThai) {

this.TrangThai = TrangThai;

}

public TreeMap<SanPham, Integer> getDSSP() {

return DSSP;

}

public void setDSSP(TreeMap<SanPham, Integer> DSSP) {

this.DSSP = DSSP;

}

}

* + - 1. **PhieuKiemChiTiet**

package model;

/\*\*

\*

\* @author Administrator

\*/

public class PhieuKiemChiTiet {

private int MaPKCT;

private String MaPhieuKiem;

private String MaSanPham;

private float DonGia;

private String DVT;

private int TonKho;

private int ThucTe;

private int SoLuongLech;

private double GiaTriLech;

public PhieuKiemChiTiet() {

}

public PhieuKiemChiTiet(int MaPKCT, String MaPhieuKiem, String MaSanPham, float DonGia, String DVT, int TonKho, int ThucTe, int SoLuongLech, double GiaTriLech) {

this.MaPKCT = MaPKCT;

this.MaPhieuKiem = MaPhieuKiem;

this.MaSanPham = MaSanPham;

this.DonGia = DonGia;

this.DVT = DVT;

this.TonKho = TonKho;

this.ThucTe = ThucTe;

this.SoLuongLech = SoLuongLech;

this.GiaTriLech = GiaTriLech;

}

public int getMaPKCT() {

return MaPKCT;

}

public void setMaPKCT(int MaPKCT) {

this.MaPKCT = MaPKCT;

}

public String getMaPhieuKiem() {

return MaPhieuKiem;

}

public void setMaPhieuKiem(String MaPhieuKiem) {

this.MaPhieuKiem = MaPhieuKiem;

}

public String getMaSanPham() {

return MaSanPham;

}

public void setMaSanPham(String MaSanPham) {

this.MaSanPham = MaSanPham;

}

public float getDonGia() {

return DonGia;

}

public void setDonGia(float DonGia) {

this.DonGia = DonGia;

}

public String getDVT() {

return DVT;

}

public void setDVT(String DVT) {

this.DVT = DVT;

}

public int getTonKho() {

return TonKho;

}

public void setTonKho(int TonKho) {

this.TonKho = TonKho;

}

public int getThucTe() {

return ThucTe;

}

public void setThucTe(int ThucTe) {

this.ThucTe = ThucTe;

}

public int getSoLuongLech() {

return SoLuongLech;

}

public void setSoLuongLech(int SoLuongLech) {

this.SoLuongLech = SoLuongLech;

}

public double getGiaTriLech() {

return GiaTriLech;

}

public void setGiaTriLech(double GiaTriLech) {

this.GiaTriLech = GiaTriLech;

}

}

* + - 1. **HoaDon**

package model;

import java.util.Date;

import java.util.TreeMap;

/\*\*

\*

\* @author Administrator

\*/

public class HoaDon {

private String MaHoaDon;

private String MaKhachHang;

private String MaNhanVien;

private Date NgayBan;

private int TongSoLuong;

private float ChietKhau;

private float VAT;

private float ThanhTien;

private float TienKhachTra;

private TreeMap<SanPham,Integer> DSSP;

public HoaDon(){

this.DSSP = new TreeMap<>();

this.ChietKhau = 0;

}

public HoaDon(String MaHoaDon, String MaKhachHang, String MaNhanVien, Date NgayBan, int TongSoLuong, float ChietKhau, float VAT, float ThanhTien, float TienKhachTra, TreeMap<SanPham,Integer> DSSP) {

this.MaHoaDon = MaHoaDon;

this.MaKhachHang = MaKhachHang;

this.MaNhanVien = MaNhanVien;

this.NgayBan = NgayBan;

this.TongSoLuong = TongSoLuong;

this.ChietKhau = ChietKhau;

this.VAT = VAT;

this.ThanhTien = ThanhTien;

this.TienKhachTra = TienKhachTra;

this.DSSP = DSSP;

}

public String getMaHoaDon() {

return MaHoaDon;

}

public void setMaHoaDon(String MaHoaDon) {

this.MaHoaDon = MaHoaDon;

}

public String getMaKhachHang() {

return MaKhachHang;

}

public void setMaKhachHang(String MaKhachHang) {

this.MaKhachHang = MaKhachHang;

}

public String getMaNhanVien() {

return MaNhanVien;

}

public void setMaNhanVien(String MaNhanVien) {

this.MaNhanVien = MaNhanVien;

}

public Date getNgayBan() {

return NgayBan;

}

public void setNgayBan(Date NgayBan) {

this.NgayBan = NgayBan;

}

public int getTongSoLuong() {

return TongSoLuong;

}

public void setTongSoLuong(int TongSoLuong) {

this.TongSoLuong = TongSoLuong;

}

public float getChietKhau() {

return ChietKhau;

}

public void setChietKhau(float ChietKhau) {

this.ChietKhau = ChietKhau;

}

public float getVAT() {

return VAT;

}

public void setVAT(float VAT) {

this.VAT = VAT;

}

public float getThanhTien() {

return ThanhTien;

}

public void setThanhTien(float ThanhTien) {

this.ThanhTien = ThanhTien;

}

public float getTienKhachTra() {

return TienKhachTra;

}

public void setTienKhachTra(float TienKhachTra) {

this.TienKhachTra = TienKhachTra;

}

public TreeMap<SanPham, Integer> getDSSP() {

return DSSP;

}

public void setDSSP(TreeMap<SanPham, Integer> DSSP) {

this.DSSP = DSSP;

}

public void themSP(SanPham SP, Integer SoLuong){

boolean chkTonTai = DSSP.containsKey(SP);

if(chkTonTai){

int SoLuongTruoc = DSSP.get(SP);

SoLuong += SoLuongTruoc;

DSSP.put(SP, SoLuong);

}

else{

DSSP.put(SP, SoLuong);

}

}

public void truSP(SanPham SP, Integer SoLuong){

boolean chkTonTai = DSSP.containsKey(SP);

if(chkTonTai){

int SoLuongTruoc = DSSP.get(SP);

SoLuong = SoLuongTruoc - SoLuong;

/\*

\* Nếu số lượng sau khi trừ xong <= 0 thì xoá sản phẩm khỏi danh sách

\* Nếu số lượng sau khi trừ xong > 0 thì xoá sản phẩm

và cập nhật lại sản phẩm cùng số lượng mới

\*/

if(SoLuong <= 0){

DSSP.remove(SP);

}

else {

DSSP.remove(SP);

DSSP.put(SP, SoLuong);

}

}

}

public void xoaSP(SanPham SP){

boolean kiemtra = DSSP.containsKey(SP);

if(kiemtra){

DSSP.remove(SP);

}

}

}

* + - 1. **HoaDonChiTiet**

package model;

/\*\*

\*

\* @author Administrator

\*/

public class HoaDonChiTiet {

private int MaHDCT;

private String MaHoaDon;

private String MaSanPham;

private float DonGia;

private int SoLuong;

private float ThanhTien;

public HoaDonChiTiet() {

}

public HoaDonChiTiet(int MaHDCT, String MaHoaDon, String MaSanPham, float DonGia, int SoLuong, float ThanhTien) {

this.MaHDCT = MaHDCT;

this.MaHoaDon = MaHoaDon;

this.MaSanPham = MaSanPham;

this.DonGia = DonGia;

this.SoLuong = SoLuong;

this.ThanhTien = ThanhTien;

}

public int getMaHDCT() {

return MaHDCT;

}

public void setMaHDCT(int MaHDCT) {

this.MaHDCT = MaHDCT;

}

public String getMaHoaDon() {

return MaHoaDon;

}

public void setMaHoaDon(String MaHoaDon) {

this.MaHoaDon = MaHoaDon;

}

public String getMaSanPham() {

return MaSanPham;

}

public void setMaSanPham(String MaSanPham) {

this.MaSanPham = MaSanPham;

}

public float getDonGia() {

return DonGia;

}

public void setDonGia(float DonGia) {

this.DonGia = DonGia;

}

public int getSoLuong() {

return SoLuong;

}

public void setSoLuong(int SoLuong) {

this.SoLuong = SoLuong;

}

public float getThanhTien() {

return ThanhTien;

}

public void setThanhTien(float ThanhTien) {

this.ThanhTien = ThanhTien;

}

}

### DAO class – Các lớp truy xuất dữ liệu

* + - 1. **DonViTinhDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import model.DonViTinh;

/\*\*

\*

\* @author Administrator

\*/

public class DonViTinhDAO {

/\*

\* Hàm tìm đơn vị tính theo mã

\*/

public DonViTinh timTheoMa(int MaDVT) {

String sql = "SELECT \* FROM DonViTinh WHERE MaDVT = ?";

List<DonViTinh> list = select(sql, MaDVT);

return list.size() > 0 ? list.get(0) : null;

}

public List<DonViTinh> select() {

String sql = "SELECT \* FROM DonViTinh";

return select(sql);

}

private List<DonViTinh> select(String sql, Object... args) {

List<DonViTinh> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

DonViTinh model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private DonViTinh readFromResultSet(ResultSet rs) throws SQLException {

DonViTinh model = new DonViTinh();

model.setMaDVT(rs.getInt("MaDVT"));

model.setMoTa(rs.getString("MoTa"));

return model;

}

}

* + - 1. **NhomHangDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import model.NhomHang;

/\*\*

\*

\* @author Administrator

\*/

public class NhomHangDAO {

public List<NhomHang> select() {

String sql = "SELECT \* FROM NhomHang";

return select(sql);

}

/\*

\* Tìm nhóm hàng theo mã nhóm hàng

\*/

public NhomHang timTheoMa(String MaNH){

String sql = "SELECT \* FROM NhomHang WHERE MaNhomHang = ?";

List<NhomHang> list = select(sql, MaNH);

return list.size() > 0 ? list.get(0) : null;

}

/\*

\* Tìm mã nhóm hàng theo tên nhóm hàng

\*/

public NhomHang timMaNhom(String TenNhom){

String sql = "SELECT \* FROM NhomHang WHERE TenNhomHang = N'" + TenNhom + "'";

List<NhomHang> list = select(sql);

return list.size() > 0 ? list.get(0) : null;

}

private List<NhomHang> select(String sql, Object... args) {

List<NhomHang> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

NhomHang model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private NhomHang readFromResultSet(ResultSet rs) throws SQLException {

NhomHang model = new NhomHang();

model.setMaNhomHang(rs.getString("MaNhomHang"));

model.setTenNhomHang(rs.getString("TenNhomHang"));

model.setGhiChu(rs.getString("GhiChu"));

return model;

}

}

* + - 1. **NhaCungCapDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import model.NhaCungCap;

/\*\*

\*

\* @author Administrator

\*/

public class NhaCungCapDAO {

/\*

\* Hàm tìm NCC theo mã (tìm tuyệt đối)

\*/

public NhaCungCap timTheoMa(String MaNCC) {

String sql = "SELECT \* FROM NhaCungCap WHERE MaNCC = ?";

List<NhaCungCap> list = select(sql, MaNCC);

return list.size() > 0 ? list.get(0) : null;

}

/\*

\* Hàm tìm NCC theo SDT

- false nếu tồn tại

- true nếu không tồn tại

\*/

public boolean kiemSDT(String SDT){

String sql = "SELECT \* FROM NhaCungCap WHERE DienThoai = ?";

List<NhaCungCap> list = select(sql, SDT);

return list.size() < 1;

}

/\*

\* Hàm tìm NhaCungCap theo Email

- false nếu tồn tại

- true nếu không tồn tại

\*/

public boolean kiemEmail(String Email){

String ChuoiDaXuLy = Email.substring(0, Email.indexOf("@"));

String sql = "SELECT \* FROM NhaCungCap WHERE Email LIKE ?";

List<NhaCungCap> list = select(sql, ChuoiDaXuLy + "%");

if(list.size() < 1)

return true;

else

return false;

}

/\*

\* Hàm thêm mới NCC

\*/

public int themNCC(NhaCungCap model) {

String sql = "INSERT INTO NhaCungCap(MaNCC, TenNCC, DienThoai, "

+ "Email, DiaChi, GhiChu, TrangThai) "

+ "VALUES (?, ?, ?, ?, ?, ?, ?)";

return jdbcHelper.executeUpdate(sql,

model.getMaNCC(),

model.getTenNCC(),

model.getDienThoai(),

model.getEmail(),

model.getDiaChi(),

model.getGhiChu(),

model.getTrangThai());

}

/\*

\* Hàm cập nhật thông tin NCC

\*/

public int capnhatNCC(NhaCungCap model) {

String sql = "UPDATE NhaCungCap SET "

+ "TenNCC = ?,DienThoai = ?, Email = ?, "

+ "DiaChi = ?, GhiChu = ?, TrangThai = ? "

+ "WHERE MaNCC = ?";

return jdbcHelper.executeUpdate(sql,

model.getTenNCC(),

model.getDienThoai(),

model.getEmail(),

model.getDiaChi(),

model.getGhiChu(),

model.getTrangThai(),

model.getMaNCC());

}

/\*

\* Hàm đếm tổng số NCC

\*/

public int tongNCC() throws SQLException{

int dem = 0;

String sql = "SELECT COUNT(\*) AS 'COUNT' FROM NhaCungCap";

ResultSet rs = jdbcHelper.executeQuery(sql);

while(rs.next()){

dem = rs.getInt(1);

}

return dem;

}

/\*

\* Hàm cập nhật trạng thái ngưng giao dịch theo mã ncc

\*/

public int ngungGiaoDich(String MaNCC) {

String sql = "UPDATE NhaCungCap SET TrangThai = 0 WHERE MaNCC = ?";

return jdbcHelper.executeUpdate(sql, MaNCC);

}

/\*

\* Hàm lọc NCC theo trạng thái

TrangThai: 0 - ngưng || 1 - hoạt động

\*/

public List<NhaCungCap> locTheoTrangThai(int TrangThai) {

String sql = "SELECT \* FROM NhaCungCap WHERE TrangThai = ? "

+ "ORDER BY CAST(RIGHT(MaNCC,LEN(MaNCC)-3) AS int)";

return select(sql, TrangThai);

}

/\*

\* Hàm tìm kiếm NCC theo mã kh, tên, sdt

\*/

public List<NhaCungCap> timTheoMaTenSDT(String ChuoiTimKiem) {

String sql = "SELECT \* FROM NhaCungCap WHERE MaNCC LIKE N'%" + ChuoiTimKiem + "%' "

+ "OR TenNCC LIKE N'%" + ChuoiTimKiem + "%' "

+ "OR DienThoai LIKE '%" + ChuoiTimKiem + "%' "

+ "ORDER BY CAST(RIGHT(MaNCC,LEN(MaNCC)-3) AS int)";

return select(sql);

}

/\*

\* Hàm tính tổng tiền nhập hàng theo Mã NCC

\*/

public double tongTienNhap(String MaNCC) throws SQLException {

double TongNhap = 0;

String sql = "SELECT SUM(ThanhTien) AS 'TongTien' FROM PhieuNhapHang "

+ "WHERE MaNCC = ? "

+ "GROUP BY MaNCC";

ResultSet rs = jdbcHelper.executeQuery(sql, MaNCC);

while(rs.next()){

TongNhap = rs.getDouble("TongTien");

}

return TongNhap;

}

/\*

\* Hàm tính tổng tiền trả hàng theo Mã NCC

\*/

public double tongTienTra(String MaNCC) throws SQLException {

double TongTra = 0;

String sql = "SELECT SUM(ThanhTien) AS 'TongTien' FROM PhieuTraHang "

+ "WHERE MaNCC = ? "

+ "GROUP BY MaNCC";

ResultSet rs = jdbcHelper.executeQuery(sql, MaNCC);

while(rs.next()){

TongTra = rs.getDouble("TongTien");

}

return TongTra;

}

public List<NhaCungCap> select() {

String sql = "SELECT \* FROM NhaCungCap "

+ "ORDER BY CAST(RIGHT(MaNCC,LEN(MaNCC)-3) AS int)";

return select(sql);

}

private List<NhaCungCap> select(String sql, Object... args) {

List<NhaCungCap> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

NhaCungCap model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private NhaCungCap readFromResultSet(ResultSet rs) throws SQLException {

NhaCungCap model = new NhaCungCap();

model.setMaNCC(rs.getString("MaNCC"));

model.setTenNCC(rs.getString("TenNCC"));

model.setDienThoai(rs.getString("DienThoai"));

model.setEmail(rs.getString("Email"));

model.setDiaChi(rs.getString("DiaChi"));

model.setGhiChu(rs.getString("GhiChu"));

model.setTrangThai(rs.getBoolean("TrangThai"));

return model;

}

}

* + - 1. **NguoiDungDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import model.NguoiDung;

/\*\*

\*

\* @author Administrator

\*/

public class NguoiDungDAO {

/\*

\* Hàm kiểm tra Email tồn tại hay chưa

true: không tồn tại

false: tồn tại

\*/

public boolean kiemEmail(String Email) {

String sql = "SELECT \* FROM NguoiDung WHERE Email = ?";

List<NguoiDung> list = select(sql, Email);

return list.size() < 1;

}

/\*

\* Hàm kiểm tra SDT tồn tại hay chưa

true: không tồn tại

false: tồn tại

\*/

public boolean kiemSDT(String SDT) {

String sql = "SELECT \* FROM NguoiDung WHERE DienThoai = ?";

List<NguoiDung> list = select(sql, SDT);

return list.size() < 1;

}

/\*

\* Hàm kiểm tra tên đăng nhập tồn tại hay chưa

true: không tồn tại

false: tồn tại

\*/

public boolean kiemTenDangNhap(String TenDangNhap) {

String sql = "SELECT \* FROM NguoiDung WHERE TenDangNhap = ?";

List<NguoiDung> list = select(sql, TenDangNhap);

return list.size() < 1;

}

/\*

\* Hàm thêm mới NguoiDung

\*/

public int themNguoiDung(NguoiDung model) {

String sql = "INSERT INTO NguoiDung(TenDangNhap, MatKhau, HoTen, DienThoai, "

+ "GioiTinh, NgaySinh, Email, DiaChi, VaiTro, TrangThai) "

+ "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

return jdbcHelper.executeUpdate(sql,

model.getTenDangNhap(),

model.getMatKhau(),

model.getHoTen(),

model.getDienThoai(),

model.getGioiTinh(),

model.getNgaySinh(),

model.getEmail(),

model.getDiaChi(),

model.getVaiTro(),

model.getTrangThai());

}

/\*

\* Hàm lọc theo trạng thái

TrangThai: 0 - ngưng || 1 - hoạt động

\*/

public List<NguoiDung> locTheoTrangThai(int TrangThai) {

String sql = "SELECT \* FROM NguoiDung WHERE TrangThai = ?";

return select(sql, TrangThai);

}

/\*

\* Hàm tìm kiếm NCC theo mã kh, tên, sdt

\*/

public List<NguoiDung> timTheoMaTenSDT(String ChuoiTimKiem) {

String sql = "SELECT \* FROM NguoiDung WHERE TenDangNhap LIKE N'%" + ChuoiTimKiem + "%' "

+ "OR HoTen LIKE N'%" + ChuoiTimKiem + "%' "

+ "OR DienThoai LIKE '%" + ChuoiTimKiem + "%'";

return select(sql);

}

/\*

\* Hàm tìm danh sách toàn bộ người dùng

\*/

public List<NguoiDung> select(){

String sql = "SELECT \* FROM NguoiDung";

return select(sql);

}

/\*

\* Hàm tìm kiếm người dùng theo Tên Đăng Nhập

\*/

public NguoiDung timTenDangNhap(String TenDangNhap){

String sql = "SELECT \* FROM NguoiDung WHERE TenDangNhap = ?";

List<NguoiDung> list = select(sql, TenDangNhap);

return list.size() > 0 ? list.get(0) : null;

}

/\*

\* Hàm cập nhật thông tin người dùng

\* @model - Model người dùng để get thông tin cập nhật

\* @check - tham số có đổi mật khẩu hay không: true có false không

\*/

public int capnhatNguoiDung(NguoiDung model, boolean check){

boolean TuyChon = check;

int rs = 0;

if(TuyChon){

String sql = "UPDATE NguoiDung SET "

+ "MatKhau = ?, "

+ "HoTen = ?, DienThoai = ?, GioiTinh = ?, NgaySinh = ?, "

+ "Email = ?, DiaChi = ?, VaiTro = ?, TrangThai = ? "

+ "WHERE TenDangNhap = ?";

return rs = jdbcHelper.executeUpdate(sql,

model.getMatKhau(),

model.getHoTen(),

model.getDienThoai(),

model.getGioiTinh(),

model.getNgaySinh(),

model.getEmail(),

model.getDiaChi(),

model.getVaiTro(),

model.getTrangThai(),

model.getTenDangNhap());

}

else {

String sql = "UPDATE NguoiDung SET "

+ "HoTen = ?, DienThoai = ?, GioiTinh = ?, NgaySinh = ?, "

+ "Email = ?, DiaChi = ?, VaiTro = ?, TrangThai = ? "

+ "WHERE TenDangNhap = ?";

return rs = jdbcHelper.executeUpdate(sql,

model.getHoTen(),

model.getDienThoai(),

model.getGioiTinh(),

model.getNgaySinh(),

model.getEmail(),

model.getDiaChi(),

model.getVaiTro(),

model.getTrangThai(),

model.getTenDangNhap());

}

}

private List<NguoiDung> select(String sql, Object... args) {

List<NguoiDung> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

NguoiDung model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private NguoiDung readFromResultSet(ResultSet rs) throws SQLException {

NguoiDung model = new NguoiDung();

model.setTenDangNhap(rs.getString("TenDangNhap"));

model.setMatKhau(rs.getString("MatKhau"));

model.setHoTen(rs.getString("HoTen"));

model.setDienThoai(rs.getString("DienThoai"));

model.setGioiTinh(rs.getBoolean("GioiTinh"));

model.setNgaySinh(rs.getDate("NgaySinh"));

model.setEmail(rs.getString("Email"));

model.setDiaChi(rs.getString("DiaChi"));

model.setVaiTro(rs.getInt("VaiTro"));

model.setTrangThai(rs.getBoolean("TrangThai"));

return model;

}

}

* + - 1. **KhachHangDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.time.LocalDate;

import java.util.ArrayList;

import java.util.List;

import model.KhachHang;

/\*\*

\*

\* @author Administrator

\*/

public class KhachHangDAO {

/\*

\* Hàm thêm mới khách hàng

\*/

public int themKhachHang(KhachHang model) {

String sql = "INSERT INTO KhachHang(MaKhachHang, HoTen, DiemThuong, DienThoai, "

+ "GioiTinh, NgaySinh, NVQuanLy, Email, DiaChi, GhiChu, TrangThai) "

+ "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

return jdbcHelper.executeUpdate(sql,

model.getMaKhachHang(),

model.getHoTen(),

model.getDiemThuong(),

model.getDienThoai(),

model.getGioiTinh(),

model.getNgaySinh(),

model.getNVQuanLy(),

model.getEmail(),

model.getDiaChi(),

model.getGhiChu(),

model.getTrangThai());

}

/\*

\* Hàm cập nhật thông tin khách hàng

\*/

public int capnhatKhachHang(KhachHang model) {

String sql = "UPDATE KhachHang SET "

+ "HoTen = ?, DiemThuong = ?, DienThoai = ?, GioiTinh = ?, "

+ "NgaySinh = ?, Email = ?, DiaChi = ?, GhiChu = ?, TrangThai = ? "

+ "WHERE MaKhachHang = ?";

return jdbcHelper.executeUpdate(sql,

model.getHoTen(),

model.getDiemThuong(),

model.getDienThoai(),

model.getGioiTinh(),

model.getNgaySinh(),

model.getEmail(),

model.getDiaChi(),

model.getGhiChu(),

model.getTrangThai(),

model.getMaKhachHang());

}

/\*

\* Hàm đếm tổng số khách hàng

\*/

public int tongKhachHang() throws SQLException{

int dem = 0;

String sql = "SELECT COUNT(\*) AS 'COUNT' FROM KhachHang";

ResultSet rs = jdbcHelper.executeQuery(sql);

while(rs.next()){

dem = rs.getInt(1);

}

return dem;

}

/\*

\* Hàm tìm kiếm khách hàng theo mã kh, tên, sdt

\*/

public List<KhachHang> timTheoMaTenSDT(String ChuoiTimKiem) {

String sql = "SELECT \* FROM KhachHang WHERE HoTen LIKE N'%" + ChuoiTimKiem + "%' "

+ "OR MaKhachHang LIKE N'%" + ChuoiTimKiem + "%' "

+ "OR DienThoai LIKE '%" + ChuoiTimKiem + "%' "

+ "ORDER BY CAST(RIGHT(MaKhachHang,LEN(MaKhachHang)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm khách hàng theo SDT

- false nếu tồn tại

- true nếu không tồn tại

\*/

public boolean kiemSDT(String SDT){

String sql = "SELECT \* FROM KhachHang WHERE DienThoai = ?";

List<KhachHang> list = select(sql, SDT);

return list.size() < 1;

}

/\*

\* Hàm tìm khách hàng theo Email

- false nếu tồn tại

- true nếu không tồn tại

\*/

public boolean kiemEmail(String Email){

String ChuoiDaXuLy = Email.substring(0, Email.indexOf("@"));

String sql = "SELECT \* FROM KhachHang WHERE Email LIKE ?";

List<KhachHang> list = select(sql, ChuoiDaXuLy + "%");

if(list.size() < 1)

return true;

else

return false;

}

/\*

\* Hàm tìm khách hàng theo mã (tìm tuyệt đối)

\*/

public KhachHang timTheoMa(String MaKhachHang) {

String sql = "SELECT \* FROM KhachHang WHERE MaKhachHang = ?";

List<KhachHang> list = select(sql, MaKhachHang);

return list.size() > 0 ? list.get(0) : null;

}

/\*

\* Hàm tìm khách hàng có sinh nhật trong tháng

\*/

public List<KhachHang> timTheoSinhNhat() {

LocalDate now = LocalDate.now();

int Thang = now.getMonthValue();

String sql = "SELECT \* FROM KhachHang WHERE MONTH(NgaySinh) = ? "

+ "ORDER BY CAST(RIGHT(MaKhachHang,LEN(MaKhachHang)-2) AS int)";

return select(sql, Thang);

}

/\*

\* Hàm tìm khách hàng theo giao dịch cuối

lọc các khách hàng có giao dịch cuối cùng vào ngày đã chọn hoặc trước đó

mục đích lọc các khách hàng quá lâu không có giao dịch nào để chuyển trạng thái

\*/

public List<KhachHang> timTheoGDCuoi(String Ngay) {

String sql = "SELECT \* FROM KhachHang WHERE NOT EXISTS "

+ "(SELECT \* FROM HoaDon WHERE NgayBan > ? "

+ "AND KhachHang.MaKhachHang = HoaDon.MaKhachHang) "

+ "ORDER BY CAST(RIGHT(KhachHang.MaKhachHang,LEN(KhachHang.MaKhachHang)-2) AS int)";

return select(sql, Ngay);

}

/\*

\* Hàm tìm khách hàng có trạng thái 'ngưng giao dịch' - TrangThai: 0 (false)

\*/

public List<KhachHang> timKHNgungHoatDong() {

String sql = "SELECT \* FROM KhachHang WHERE TrangThai = 0 "

+ "ORDER BY CAST(RIGHT(MaKhachHang,LEN(MaKhachHang)-2) AS int)";

return select(sql);

}

/\*

\* Hàm cập nhật trạng thái ngưng giao dịch cho khách hàng theo mã

\*/

public int ngungGiaoDich(String MaKH) {

String sql = "UPDATE KhachHang SET TrangThai = 0 WHERE MaKhachHang = ?";

return jdbcHelper.executeUpdate(sql, MaKH);

}

public List<KhachHang> select() {

String sql = "SELECT \* FROM KhachHang "

+ "ORDER BY CAST(RIGHT(MaKhachHang,LEN(MaKhachHang)-2) AS int)";

return select(sql);

}

public List<KhachHang> timKhachHang(String ChuoiTimKiem) {

String sql = "SELECT \* FROM KhachHang WHERE HoTen LIKE N'%" + ChuoiTimKiem + "%' "

+ "OR DienThoai LIKE N'%" + ChuoiTimKiem + "%' "

+ "ORDER BY CAST(RIGHT(MaKhachHang,LEN(MaKhachHang)-2) AS int)";

return select(sql);

}

public KhachHang timKhachHangTheoMa(String MaKhachHang) {

String sql = "SELECT \* FROM KhachHang WHERE MaKhachHang = ?";

List<KhachHang> listKH = select(sql, MaKhachHang);

return listKH.get(0);

}

/\*

\* Hàm cập nhật điểm thưởng khách hàng sau khi áp dụng chiết khấu

\*/

public void capnhatDiemThuong(KhachHang model){

String sql = "UPDATE KhachHang SET DiemThuong = ? WHERE MaKhachHang = ?";

jdbcHelper.executeUpdate(sql, model.getDiemThuong(), model.getMaKhachHang());

}

private List<KhachHang> select(String sql, Object... args) {

List<KhachHang> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

KhachHang model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private KhachHang readFromResultSet(ResultSet rs) throws SQLException {

KhachHang model = new KhachHang();

model.setMaKhachHang(rs.getString("MaKhachHang"));

model.setHoTen(rs.getString("HoTen"));

model.setDiemThuong(rs.getInt("DiemThuong"));

model.setDienThoai(rs.getString("DienThoai"));

model.setGioiTinh(rs.getBoolean("GioiTinh"));

model.setNgaySinh(rs.getDate("NgaySinh"));

model.setNVQuanLy(rs.getString("NVQuanLy"));

model.setEmail(rs.getString("Email"));

model.setDiaChi(rs.getString("DiaChi"));

model.setGhiChu(rs.getString("GhiChu"));

model.setTrangThai(rs.getBoolean("TrangThai"));

return model;

}

}

* + - 1. **SanPhamDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.Date;

import java.util.List;

import model.SanPham;

/\*\*

\*

\* @author Administrator

\*/

public class SanPhamDAO {

/\*

\* Hàm lọc sản phẩm để bán

- trạng thái: đang kinh doanh

- tồn kho: > 0

\*/

public List<SanPham> locSanPhamDangBan() {

String sql = "SELECT \* FROM SanPham WHERE TrangThai = 1 AND TonKho > 0";

return select(sql);

}

/\*

\* Hàm lọc sp đã hết HSD

\*/

public List<SanPham> locHetHSD(String MaNH, boolean TrangThai) {

String sql = "SELECT \* FROM SanPham WHERE HSDTruoc < getdate()";

if (!MaNH.equalsIgnoreCase("0"))

sql += " AND MaNhomHang = '" + MaNH + "'";

if (!TrangThai)

sql += " AND TrangThai = 0";

sql += " ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc HSD 3 ngày tới

\*/

public List<SanPham> locHSD3NgayToi(String MaNH, boolean TrangThai) {

String sql = "SELECT \* FROM SanPham WHERE HSDTruoc <= (getdate()+3) "

+ "AND HSDTruoc >= getdate()";

if (!MaNH.equalsIgnoreCase("0"))

sql += " AND MaNhomHang = '" + MaNH + "'";

if (!TrangThai)

sql += " AND TrangThai = 0";

sql += " ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc HSD 7 ngày tới

\*/

public List<SanPham> locHSD7NgayToi(String MaNH, boolean TrangThai) {

String sql = "SELECT \* FROM SanPham WHERE HSDTruoc <= (getdate()+7) "

+ "AND HSDTruoc >= getdate()";

if (!MaNH.equalsIgnoreCase("0"))

sql += " AND MaNhomHang = '" + MaNH + "'";

if (!TrangThai)

sql += " AND TrangThai = 0";

sql += " ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc sp hết HSD trong tháng này

\*/

public List<SanPham> locHSDThangNay(String MaNH, boolean TrangThai) {

String sql = "SELECT \* FROM SanPham WHERE "

+ "YEAR(HSDTruoc) = YEAR(getdate()) "

+ "AND MONTH(HSDTruoc) = MONTH(getdate()) "

+ "AND HSDTruoc >= getdate()";

if (!MaNH.equalsIgnoreCase("0"))

sql += " AND MaNhomHang = '" + MaNH + "'";

if (!TrangThai)

sql += " AND TrangThai = 0";

sql += " ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc sp hết HSD trong tháng tới

\*/

public List<SanPham> locHSDThangToi(String MaNH, boolean TrangThai) {

String sql = "SELECT \* FROM SanPham WHERE "

+ "YEAR(HSDTruoc) = YEAR(dateadd(MONTH, 1, getdate())) "

+ "AND MONTH(HSDTruoc) = MONTH(dateadd(MONTH, 1, getdate()))";

if (!MaNH.equalsIgnoreCase("0"))

sql += " AND MaNhomHang = '" + MaNH + "'";

if (!TrangThai)

sql += " AND TrangThai = 0";

sql += " ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Hàm thêm mới SP

\*/

public int themSanPham(SanPham model) {

String sql = "INSERT INTO SanPham(MaSanPham, TenSanPham, MaVach, MaNhomHang, "

+ "MaNCC, MaDVT, TonKho, DinhMucNhoNhat, DinhMucLonNhat, GiaVon, "

+ "GiaBan, HSDTruoc, HSDMoiNhat, HinhAnh, TrangThai) "

+ "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?)";

return jdbcHelper.executeUpdate(sql,

model.getMaSanPham(),

model.getTenSanPham(),

model.getMaVach(),

model.getMaNhomHang(),

model.getMaNCC(),

model.getMaDVT(),

model.getTonKho(),

model.getDinhMucNhoNhat(),

model.getDinhMucLonNhat(),

model.getGiaVon(),

model.getGiaBan(),

model.getHSDTruoc(),

model.getHSDMoiNhat(),

model.getHinhAnh(),

model.getTrangThai());

}

/\*

\* Hàm cập nhật tồn kho sản phẩm

- sử dụng kèm nghiệp vụ xuất - nhập - huỷ - kiểm kê

\*/

public int capnhatTonKho(String MaSP, int TonKho) {

String sql = "UPDATE SanPham SET TonKho = ? WHERE MaSanPham = ?";

return jdbcHelper.executeUpdate(sql, TonKho, MaSP);

}

/\*

\* Hàm cập nhật giá vốn sản phẩm

- sử dụng kèm nghiệp vụ xuất - nhập - huỷ - kiểm kê

\*/

public int capnhatGiaVon(String MaSP, float GiaVonMoi) {

String sql = "UPDATE SanPham SET GiaVon = ? WHERE MaSanPham = ?";

return jdbcHelper.executeUpdate(sql, GiaVonMoi, MaSP);

}

/\*

\* Hàm cập nhật HSD Mới nhất cho sản phẩm

- sử dụng kèm nghiệp vụ xuất - nhập - huỷ - kiểm kê

\*/

public int capnhatHSDMoi(String MaSP, Date HSDMoiNhat) {

String sql = "UPDATE SanPham SET HSDMoiNhat = ? WHERE MaSanPham = ?";

return jdbcHelper.executeUpdate(sql, HSDMoiNhat, MaSP);

}

/\*

\* Hàm cập nhật thông tin San Pham

- không cập nhật giá vốn (vì tính theo công thức riêng)

- không cập nhật tồn kho (phụ thuộc nghiệp vụ nhập - xuất - kiểm kho)

\*/

public int capnhatSP(SanPham model) {

String sql = "UPDATE SanPham SET "

+ "TenSanPham = ?, MaVach = ?, MaNhomHang = ?, "

+ "MaNCC = ?, MaDVT = ?, DinhMucNhoNhat = ?, "

+ "DinhMucLonNhat = ?, GiaBan = ?, HSDTruoc = ?, "

+ "HSDMoiNhat = ?, HinhAnh = ?, TrangThai = ? "

+ "WHERE MaSanPham = ?";

return jdbcHelper.executeUpdate(sql,

model.getTenSanPham(),

model.getMaVach(),

model.getMaNhomHang(),

model.getMaNCC(),

model.getMaDVT(),

model.getDinhMucNhoNhat(),

model.getDinhMucLonNhat(),

model.getGiaBan(),

model.getHSDTruoc(),

model.getHSDMoiNhat(),

model.getHinhAnh(),

model.getTrangThai(),

model.getMaSanPham());

}

/\*

\* Hàm đếm tổng số sản phẩm

\*/

public int demSanPham() throws SQLException {

int dem = 0;

String sql = "SELECT COUNT(\*) AS 'COUNT' FROM SanPham";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

dem = rs.getInt("COUNT");

}

return dem;

}

/\*

\* Hàm chuyển trạng thái ngừng kinh doanh

\*/

public int ngungKinhDoanh(String MaSP) {

String sql = "UPDATE SanPham SET TrangThai = 0 "

+ "WHERE MaSanPham = ?";

return jdbcHelper.executeUpdate(sql, MaSP);

}

/\*

\* Hàm tìm sản phẩm theo mã Nhà cung cấp

\*/

public List<SanPham> timTheoNCC(String MaNCC) {

String sql = "SELECT \* FROM SanPham WHERE MaNCC = ? "

+ "ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql, MaNCC);

}

/\*

\* Hàm tìm sản phẩm theo mã SP hoặc tên SP

\*/

public List<SanPham> timTheoMaHoacTen(String ChuoiTimKiem) {

String sql = "SELECT \* FROM SanPham WHERE MaSanPham LIKE '" + ChuoiTimKiem + "' "

+ "OR TenSanPham LIKE N'%" + ChuoiTimKiem + "%' "

+ "ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc sản phẩm ngừng kinh doanh

\*/

public List<SanPham> locNgungKinhDoanh() {

String sql = "SELECT \* FROM SanPham WHERE TrangThai = 0 "

+ "ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc sản phẩm đang kinh doanh

\*/

public List<SanPham> locDangKinhDoanh() {

String sql = "SELECT \* FROM SanPham WHERE TrangThai = 1 "

+ "ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc sản phẩm hết hàng trong kho

\*/

public List<SanPham> locHetHang() {

String sql = "SELECT \* FROM SanPham WHERE TonKho = 0 "

+ "ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc sản phẩm còn hàng trong kho

\*/

public List<SanPham> locConHang() {

String sql = "SELECT \* FROM SanPham WHERE TonKho > 0 "

+ "ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc sản phẩm vượt định mức tồn kho

\*/

public List<SanPham> locVuotDinhMuc() {

String sql = "SELECT \* FROM SanPham WHERE TonKho > DinhMucLonNhat "

+ "ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc sản phẩm dưới định mức tồn kho

\*/

public List<SanPham> locDuoiDinhMuc() {

String sql = "SELECT \* FROM SanPham WHERE TonKho < DinhMucNhoNhat "

+ "ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

public List<SanPham> select() {

String sql = "SELECT \* FROM SanPham "

+ "ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Tìm kiếm sản phẩm chính xác theo Mã Vạch

\* Kêt quả trả về: 01 sản phẩm

\*/

public SanPham timTheoMaVach(String MaVach){

String sql = "SELECT \* FROM SanPham WHERE MaVach = ?";

List<SanPham> list = select(sql, MaVach);

return list.size() > 0 ? list.get(0) : null;

}

/\*

\* Tìm kiếm sản phẩm chính xác theo Mã SP

\* Kêt quả trả về: 01 sản phẩm

\*/

public SanPham timTheoMaSP(String MaSP){

String sql = "SELECT \* FROM SanPham WHERE MaSanPham = ?";

List<SanPham> list = select(sql, MaSP);

return list.size() > 0 ? list.get(0) : null;

}

/\*

\* Tìm kiếm tên sản phẩm có điều kiện theo nhóm hàng

\* Kết quả trả về là list các sản phẩm (tìm kiếm tương đối)

\*/

public List<SanPham> timSanPhamTheoTen(String ChuoiTimKiem){

String sql = "SELECT \* FROM SanPham WHERE TenSanPham LIKE N'%" + ChuoiTimKiem + "%' "

+ "ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql);

}

/\*

\* Lọc sản phẩm theo mã nhóm hàng

\* Kết quả trả về là list các sản phẩm trong nhóm hàng đó

\*/

public List<SanPham> timTheoMaNhom(String MaNhomHang){

String sql = "SELECT \* FROM SanPham WHERE MaNhomHang LIKE ? "

+ "ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

return select(sql, MaNhomHang);

}

/\*

\* Tìm kiếm tên sản phẩm có điều kiện theo nhóm hàng

\* Kết quả trả về là list các sản phẩm (tìm kiếm tương đối)

\*/

public List<SanPham> timKetHopNhomHang(String ChuoiTimKiem, String MaNhomHang){

String sql = "SELECT \* FROM SanPham WHERE TenSanPham LIKE N'%" + ChuoiTimKiem + "%' "

+ "AND MaNhomHang LIKE '" + MaNhomHang + "' "

+ "ORDER BY CAST(RIGHT(MaSanPham,LEN(MaSanPham)-2) AS int)";

List<SanPham> list = select(sql);

return list.size() > 0 ? list : null;

}

private List<SanPham> select(String sql, Object... args) {

List<SanPham> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

SanPham model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private SanPham readFromResultSet(ResultSet rs) throws SQLException {

SanPham model = new SanPham();

model.setMaSanPham(rs.getString("MaSanPham"));

model.setTenSanPham(rs.getString("TenSanPham"));

model.setMaVach(rs.getString("MaVach"));

model.setMaNhomHang(rs.getString("MaNhomHang"));

model.setMaNCC(rs.getString("MaNCC"));

model.setMaDVT(rs.getInt("MaDVT"));

model.setTonKho(rs.getInt("TonKho"));

model.setDinhMucNhoNhat(rs.getInt("DinhMucNhoNhat"));

model.setDinhMucLonNhat(rs.getInt("DinhMucLonNhat"));

model.setGiaVon(rs.getFloat("GiaVon"));

model.setGiaBan(rs.getFloat("GiaBan"));

model.setHSDTruoc(rs.getDate("HSDTruoc"));

model.setHSDMoiNhat(rs.getDate("HSDMoiNhat"));

model.setHinhAnh(rs.getString("HinhAnh"));

model.setTrangThai(rs.getBoolean("TrangThai"));

return model;

}

}

* + - 1. **HoaDonDAO**

package dao;

import helper.DateHelper;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import model.HoaDon;

/\*\*

\*

\* @author Administrator

\*/

public class HoaDonDAO {

/\*

\* Hàm lọc số hoá đơn bán ra trong năm

\*/

public List<HoaDon> locHoaDonTrongNam() {

String sql = "SELECT \* FROM HoaDon "

+ "WHERE YEAR(NgayBan) = YEAR(getdate()) "

+ "ORDER BY CAST(RIGHT(MaHoaDon,LEN(MaHoaDon)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc số hoá đơn bán ra trong quý

\*/

public List<HoaDon> locHoaDonTrongQuy() {

int Quy = DateHelper.tinhSoQuyTrongNam();

String sql = "SELECT \* FROM HoaDon "

+ "WHERE YEAR(NgayBan) = YEAR(getdate()) ";

switch (Quy) {

case 1:

sql+= "AND MONTH(NgayBan) >= 1 AND MONTH(NgayBan) <= 3";

break;

case 2:

sql+= "AND MONTH(NgayBan) >= 4 AND MONTH(NgayBan) <= 6";

break;

case 3:

sql+= "AND MONTH(NgayBan) >= 7 AND MONTH(NgayBan) <= 9";

break;

case 4:

sql+= "AND MONTH(NgayBan) >= 10 AND MONTH(NgayBan) <= 12";

break;

}

sql += " ORDER BY CAST(RIGHT(MaHoaDon,LEN(MaHoaDon)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc số hoá đơn bán ra trong tháng

\*/

public List<HoaDon> locHoaDonTrongThang() {

String sql = "SELECT \* FROM HoaDon "

+ "WHERE MONTH(NgayBan) = MONTH(getdate()) "

+ "AND YEAR(NgayBan) = YEAR(getdate()) "

+ "ORDER BY CAST(RIGHT(MaHoaDon,LEN(MaHoaDon)-2) AS int)";

return select(sql);

}

/\*

\* Hàm lọc số hoá đơn bán ra trong tuần

\*/

public List<HoaDon> locHoaDonTrongTuan() {

String DauTuan = DateHelper.tinhNgayDauTuan(1);

String sql = "SELECT \* FROM HoaDon "

+ "WHERE NgayBan <= FORMAT(getdate(), 'yyyy-MM-dd') "

+ "AND NgayBan >= '" + DauTuan + "' "

+ "ORDER BY CAST(RIGHT(MaHoaDon,LEN(MaHoaDon)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tính thành tiền hoá đơn (chưa gồm VAT và chiết khấu)

\*/

public float tinhThanhTien(String MaHD) throws SQLException {

float ThanhTien = 0;

String sql = "SELECT SUM(ThanhTien) AS 'ThanhTien' FROM HoaDonChiTiet "

+ "WHERE MaHoaDon = ? "

+ "GROUP BY MaHoaDon";

ResultSet rs = jdbcHelper.executeQuery(sql, MaHD);

while (rs.next()) {

ThanhTien = rs.getFloat("ThanhTien");

}

return ThanhTien;

}

/\*

\* Hàm tìm hoá đơn theo khoảng ngày

\*/

public List<HoaDon> timTheoKhoangNgay(String TuNgay, String DenNgay) {

String sql = "SELECT \* FROM HoaDon "

+ "WHERE NgayBan >= '" + TuNgay + "' "

+ "AND NgayBan <= '" + DenNgay + "' "

+ "ORDER BY CAST(RIGHT(MaHoaDon,LEN(MaHoaDon)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm hoá đơn tạo trong 7 ngày trước (không tính hôm nay)

\*/

public List<HoaDon> timTrong7NgayTruoc() {

String sql = "SELECT \* FROM HoaDon "

+ "WHERE NgayBan >= FORMAT(getdate()-7,'yyyy-MM-dd') "

+ "AND NgayBan < FORMAT(getdate(),'yyyy-MM-dd') "

+ "ORDER BY CAST(RIGHT(MaHoaDon,LEN(MaHoaDon)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm hoá đơn tạo trong hôm qua

\*/

public List<HoaDon> timTrongHomQua() {

String sql = "SELECT \* FROM HoaDon "

+ "WHERE NgayBan = FORMAT(getdate()-1,'yyyy-MM-dd') "

+ "ORDER BY CAST(RIGHT(MaHoaDon,LEN(MaHoaDon)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm hoá đơn tạo trong ngày

\*/

public List<HoaDon> timTrongNgay() {

String sql = "SELECT \* FROM HoaDon "

+ "WHERE NgayBan = FORMAT(getdate(),'yyyy-MM-dd') "

+ "ORDER BY CAST(RIGHT(MaHoaDon,LEN(MaHoaDon)-2) AS int)";

return select(sql);

}

/\*

\* Hàm gọi tất cả các hoá đơn

\*/

public List<HoaDon> select() {

String sql = "SELECT \* FROM HoaDon "

+ "ORDER BY CAST(RIGHT(MaHoaDon,LEN(MaHoaDon)-2) AS int)";

return select(sql);

}

/\*

\* Insert Hoá đơn bằng Store Procedure

\*/

public int themHoaDon(HoaDon model) {

int rs = 0;

String sql = "{call sp\_insert\_hoadon(?, ?, ?, ?, ?, ?, ?, ?)}";

try {

rs = jdbcHelper.executeUpdate(sql,

model.getMaKhachHang(),

model.getMaNhanVien(),

DateHelper.toString(model.getNgayBan(), "yyyy-MM-dd"),

model.getTongSoLuong(),

model.getChietKhau(),

model.getVAT(),

model.getThanhTien(),

model.getTienKhachTra());

} catch (Exception e) {

e.printStackTrace();

}

return rs;

}

/\*

\* Hàm đếm tổng số hoá đơn

\*/

public int demHoaDon() throws SQLException{

int dem = 0;

String sql = "SELECT COUNT(\*) AS 'COUNT' FROM HoaDon";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

dem = rs.getInt("COUNT");

}

return dem;

}

/\*

\* Hàm đếm số hoá đơn đã xuất của nhân viên trong ngày

\*/

public int demHoaDonTrongNgay(String MaNhanVien) throws SQLException{

int dem = 0;

String NgayBan = DateHelper.toString(DateHelper.now(), "yyyy-MM-dd");

String sql = "SELECT COUNT(\*) AS 'COUNT' FROM HoaDon "

+ "WHERE MaNhanVien = ? AND NgayBan = ?";

ResultSet rs = jdbcHelper.executeQuery(sql, MaNhanVien, NgayBan);

while (rs.next()) {

dem = rs.getInt("COUNT");

}

return dem;

}

/\*

\* Hàm thống kê doanh thu trong ngày của nhân viên

\*/

public float tinhDoanhThuTrongNgay(String MaNhanVien) throws SQLException{

float DoanhThu = 0;

String NgayBan = DateHelper.toString(DateHelper.now(), "yyyy-MM-dd");

String sql = "SELECT SUM(ThanhTien) AS 'DoanhThu' FROM HoaDon "

+ "WHERE MaNhanVien = ? AND NgayBan = ? "

+ "GROUP BY MaNhanVien";

ResultSet rs = jdbcHelper.executeQuery(sql, MaNhanVien, NgayBan);

while (rs.next()) {

DoanhThu = rs.getFloat("DoanhThu");

}

return DoanhThu;

}

/\*

\* Hàm lọc danh sách các đơn hàng nhân viên đã bán trong ngày

\*/

public List<HoaDon> timDonHangTrongNgay(String MaNhanVien){

String sql = "SELECT \* FROM HoaDon WHERE MaNhanVien = ? AND NgayBan = ? "

+ "ORDER BY CAST(RIGHT(MaHoaDon,LEN(MaHoaDon)-2) AS int)";

String NgayBan = DateHelper.toString(DateHelper.now(), "yyyy-MM-dd");

return select(sql, MaNhanVien, NgayBan);

}

/\*

\* Hàm tìm hoá đơn theo mã người bán

\*/

public List<HoaDon> timHoaDonTheoMaNguoiBan(String MaNguoiBan){

String sql = "SELECT \* FROM HoaDon WHERE MaNhanVien = ?";

return select(sql, MaNguoiBan);

}

/\*

\* Hàm tìm hoá đơn theo tên khách hàng

\*/

public List<HoaDon> timHoaDonTheoTenKH(String TenKhachHang){

String sql = "SELECT \* FROM HoaDon WHERE EXISTS "

+ "(SELECT \* FROM KhachHang WHERE HoTen LIKE N'%" + TenKhachHang + "%' "

+ "AND HoaDon.MaKhachHang = KhachHang.MaKhachHang) "

+ "ORDER BY CAST(RIGHT(MaHoaDon,LEN(MaHoaDon)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm hoá đơn theo mã

\*/

public HoaDon timHoaDonTheoMa(String MaHoaDon){

String sql = "SELECT \* FROM HoaDon WHERE MaHoaDon = ?";

List<HoaDon> list = select(sql, MaHoaDon);

return list.size() > 0 ? list.get(0) : null;

}

private List<HoaDon> select(String sql, Object... args) {

List<HoaDon> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

HoaDon model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private HoaDon readFromResultSet(ResultSet rs) throws SQLException {

HoaDon model = new HoaDon();

model.setMaHoaDon(rs.getString("MaHoaDon"));

model.setMaKhachHang(rs.getString("MaKhachHang"));

model.setMaNhanVien(rs.getString("MaNhanVien"));

model.setNgayBan(rs.getDate("NgayBan"));

model.setTongSoLuong(rs.getInt("TongSoLuong"));

model.setChietKhau(rs.getFloat("ChietKhau"));

model.setVAT(rs.getFloat("VAT"));

model.setThanhTien(rs.getFloat("ThanhTien"));

model.setTienKhachTra(rs.getFloat("TienKhachTra"));

return model;

}}

* + - 1. **HoaDonChiTietDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import model.HoaDonChiTiet;

/\*\*

\*

\* @author Administrator

\*/

public class HoaDonChiTietDAO {

/\*

\* Insert Hoá đơn chi tiết

\*/

public void themHDCT(HoaDonChiTiet model){

String sql = "INSERT INTO HoaDonChiTiet(MaHoaDon, MaSanPham, DonGia, "

+ "SoLuong, ThanhTien) "

+ "VALUES (?, ?, ?, ?, ?)";

jdbcHelper.executeUpdate(sql,

model.getMaHoaDon(),

model.getMaSanPham(),

model.getDonGia(),

model.getSoLuong(),

model.getThanhTien());

}

/\*

\* Tìm danh sách hoá đơn chi tiết theo mã hoá đơn

\*/

public List<HoaDonChiTiet> timDanhSachHDCT(String MaHoaDon){

String sql = "SELECT \* FROM HoaDonChiTiet WHERE MaHoaDon = ?";

return select(sql, MaHoaDon);

}

private List<HoaDonChiTiet> select(String sql, Object... args) {

List<HoaDonChiTiet> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

HoaDonChiTiet model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private HoaDonChiTiet readFromResultSet(ResultSet rs) throws SQLException {

HoaDonChiTiet model = new HoaDonChiTiet();

model.setMaHDCT(rs.getInt("MaHDCT"));

model.setMaHoaDon(rs.getString("MaHoaDon"));

model.setMaSanPham(rs.getString("MaSanPham"));

model.setDonGia(rs.getFloat("DonGia"));

model.setSoLuong(rs.getInt("SoLuong"));

model.setThanhTien(rs.getFloat("ThanhTien"));

return model;

}}

* + - 1. **PhieuNhapDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import model.PhieuNhap;

/\*\*

\*

\* @author Administrator

\*/

public class PhieuNhapDAO {

/\*

\* Hàm cập nhật phiếu nhập hàng

chỉ được cập nhật ghi chú

\*/

public int capnhatPhieuNhap(PhieuNhap model) {

String sql = "UPDATE PhieuNhapHang SET GhiChu = ? "

+ "WHERE MaPhieuNhap = ?";

return jdbcHelper.executeUpdate(sql,

model.getGhiChu(),

model.getMaPhieuNhap());

}

/\*

\* Hàm thêm phiếu nhập

\*/

public int themPhieuNhap(PhieuNhap model) {

String sql = "INSERT INTO PhieuNhapHang(MaPhieuNhap, MaNhanVien, NgayNhap, "

+ "MaNCC, TongSoLuong, ChietKhau, VAT, ThanhTien, GhiChu) "

+ "VALUES (?, ?, ?, ?, ?, ?, ?, ?, ?)";

return jdbcHelper.executeUpdate(sql,

model.getMaPhieuNhap(),

model.getMaNhanVien(),

model.getNgayNhap(),

model.getMaNCC(),

model.getTongSoLuong(),

model.getChietKhau(),

model.getVAT(),

model.getThanhTien(),

model.getGhiChu());

}

/\*

\* Hàm đếm tổng số phiếu nhập

\*/

public int demPhieuNhap() throws SQLException {

int dem = 0;

String sql = "SELECT COUNT(\*) AS 'COUNT' FROM PhieuNhapHang";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next())

dem = rs.getInt("COUNT");

return dem;

}

/\*

\* Hàm tìm phiếu nhập theo ngày tuỳ chọn

\*/

public List<PhieuNhap> timTheoKhoangNgay(String TuNgay, String DenNgay) {

String sql = "SELECT \* FROM PhieuNhapHang "

+ "WHERE NgayNhap >= '" + TuNgay + "' "

+ "AND NgayNhap <= '" + DenNgay + "' "

+ "ORDER BY CAST(RIGHT(MaPhieuNhap, LEN(MaPhieuNhap)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm phiếu nhập trong 7 ngày trước (không tính hôm nay)

\*/

public List<PhieuNhap> tim7NgayTruoc() {

String sql = "SELECT \* FROM PhieuNhapHang "

+ "WHERE NgayNhap < FORMAT(getdate(),'yyyy-MM-dd') "

+ "AND NgayNhap >= FORMAT(getdate()-7,'yyyy-MM-dd')"

+ "ORDER BY CAST(RIGHT(MaPhieuNhap, LEN(MaPhieuNhap)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm phiếu nhập tạo hôm qua

\*/

public List<PhieuNhap> timNgayHomQua() {

String sql = "SELECT \* FROM PhieuNhapHang "

+ "WHERE NgayNhap = FORMAT(getdate()-1,'yyyy-MM-dd') "

+ "ORDER BY CAST(RIGHT(MaPhieuNhap, LEN(MaPhieuNhap)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm phiếu nhập tạo trong ngày

\*/

public List<PhieuNhap> timTrongNgay() {

String sql = "SELECT \* FROM PhieuNhapHang "

+ "WHERE NgayNhap = FORMAT(getdate(),'yyyy-MM-dd') "

+ "ORDER BY CAST(RIGHT(MaPhieuNhap, LEN(MaPhieuNhap)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm theo mã sản phẩm hoặc tên sản phẩm

\*/

public List<PhieuNhap> timTheoSanPham(String ChuoiTimKiem) {

String sql = "SELECT \* FROM PhieuNhapHang PN "

+ "WHERE EXISTS "

+ "(SELECT MaPhieuNhap FROM PhieuNhapChiTiet PNCT "

+ "WHERE PNCT.MaPhieuNhap = PN.MaPhieuNhap "

+ "AND EXISTS "

+ "(SELECT MaSanPham FROM SanPham SP "

+ "WHERE PNCT.MaSanPham = SP.MaSanPham "

+ "AND (MaSanPham = '" + ChuoiTimKiem +"' OR TenSanPham LIKE N'%" + ChuoiTimKiem + "%'))) "

+ "ORDER BY CAST(RIGHT(MaPhieuNhap, LEN(MaPhieuNhap)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm theo mã NCC

\*/

public List<PhieuNhap> timTheoMaNCC(String MaNCC){

String sql = "SELECT \* FROM PhieuNhapHang WHERE MaNCC = ?";

return select(sql, MaNCC);

}

/\*

\* Hàm tìm theo mã chứng từ

\*/

public PhieuNhap timTheoMa(String MaChungTu){

String sql = "SELECT \* FROM PhieuNhapHang WHERE MaPhieuNhap = ?";

List<PhieuNhap> list = select(sql, MaChungTu);

return list.size() > 0 ? list.get(0) : null;

}

/\*

\* Hàm tìm tất cả phiếu nhập

\*/

public List<PhieuNhap> select(){

String sql = "SELECT \* FROM PhieuNhapHang "

+ "ORDER BY CAST(RIGHT(MaPhieuNhap, LEN(MaPhieuNhap)-2) AS int)";

return select(sql);

}

private List<PhieuNhap> select(String sql, Object... args) {

List<PhieuNhap> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

PhieuNhap model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private PhieuNhap readFromResultSet(ResultSet rs) throws SQLException {

PhieuNhap model = new PhieuNhap();

model.setMaPhieuNhap(rs.getString("MaPhieuNhap"));

model.setMaNhanVien(rs.getString("MaNhanVien"));

model.setNgayNhap(rs.getDate("NgayNhap"));

model.setMaNCC(rs.getString("MaNCC"));

model.setTongSoLuong(rs.getInt("TongSoLuong"));

model.setChietKhau(rs.getFloat("ChietKhau"));

model.setVAT(rs.getFloat("VAT"));

model.setThanhTien(rs.getDouble("ThanhTien"));

model.setGhiChu(rs.getString("GhiChu"));

return model;

}}

* + - 1. **PhieuNhapChiTietDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import model.PhieuNhapChiTiet;

/\*\*

\*

\* @author Administrator

\*/

public class PhieuNhapChiTietDAO {

/\*

\* Insert phiếu nhập chi tiết

\*/

public void themPNCT(PhieuNhapChiTiet model){

String sql = "INSERT INTO PhieuNhapChiTiet(MaPhieuNhap, MaSanPham, SoLuong, "

+ "GiaNhap, ThanhTien, HSDMoiNhat) "

+ "VALUES (?, ?, ?, ?, ?, ?)";

jdbcHelper.executeUpdate(sql,

model.getMaPhieuNhap(),

model.getMaSanPham(),

model.getSoLuong(),

model.getGiaNhap(),

model.getThanhTien(),

model.getHSDMoiNhat());

}

/\*

\* Tìm danh sách phiếu nhập chi tiết theo mã phiếu nhập

\*/

public List<PhieuNhapChiTiet> timDanhSachPNCT(String MaPhieuNhap){

String sql = "SELECT \* FROM PhieuNhapChiTiet WHERE MaPhieuNhap = ?";

return select(sql, MaPhieuNhap);

}

private List<PhieuNhapChiTiet> select(String sql, Object... args) {

List<PhieuNhapChiTiet> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

PhieuNhapChiTiet model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private PhieuNhapChiTiet readFromResultSet(ResultSet rs) throws SQLException {

PhieuNhapChiTiet model = new PhieuNhapChiTiet();

model.setMaPNCT(rs.getInt("MaPNCT"));

model.setMaPhieuNhap(rs.getString("MaPhieuNhap"));

model.setMaSanPham(rs.getString("MaSanPham"));

model.setSoLuong(rs.getInt("SoLuong"));

model.setGiaNhap(rs.getFloat("GiaNhap"));

model.setThanhTien(rs.getDouble("ThanhTien"));

model.setHSDMoiNhat(rs.getDate("HSDMoiNhat"));

return model;

}}

* + - 1. **PhieuHuyDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import model.PhieuHuyHang;

/\*\*

\*

\* @author Administrator

\*/

public class PhieuHuyDAO {

/\*

\* Hàm thêm phiếu huỷ

\*/

public int themPhieuHuy(PhieuHuyHang model) {

String sql = "INSERT INTO PhieuHuyHang(MaPhieuHuy, LoaiPhieuHuy, MaNhanVien, "

+ "NgayHuy, SoLuongHuy, GiaTriHuy, GhiChu) "

+ "VALUES (?, ?, ?, ?, ?, ?, ?)";

return jdbcHelper.executeUpdate(sql,

model.getMaPhieuHuy(),

model.getLoaiPhieuHuy(),

model.getMaNhanVien(),

model.getNgayHuy(),

model.getSoLuongHuy(),

model.getGiaTriHuy(),

model.getGhiChu()

);

}

/\*

\* Hàm tìm phiếu theo loại phiếu

\*/

public List<PhieuHuyHang> timTheoLoaiPhieu(String LoaiPhieuHuy) {

char x[] = LoaiPhieuHuy.toCharArray();

String LoaiPhieu = "";

for(int i = 0; i < x.length; i++){

if (i != x.length - 1)

LoaiPhieu += x[i] + ",";

else

LoaiPhieu += x[i];

}

String sql = "SELECT \* FROM PhieuHuyHang "

+ "WHERE LoaiPhieuHuy IN (" + LoaiPhieu + ") "

+ "ORDER BY CAST(RIGHT(MaPhieuHuy,LEN(MaPhieuHuy)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm phiếu huỷ theo ngày tuỳ chọn

\*/

public List<PhieuHuyHang> timTheoKhoangNgay(String LoaiPhieuKiem, String TuNgay, String DenNgay) {

char x[] = LoaiPhieuKiem.toCharArray();

String LoaiPhieu = "";

for(int i = 0; i < x.length; i++){

if (i != x.length - 1)

LoaiPhieu += x[i] + ",";

else

LoaiPhieu += x[i];

}

String sql = "SELECT \* FROM PhieuHuyHang "

+ "WHERE LoaiPhieuHuy IN (" + LoaiPhieu + ") "

+ "AND NgayHuy >= '" + TuNgay + "' "

+ "AND NgayHuy <= '" + DenNgay + "' "

+ "ORDER BY CAST(RIGHT(MaPhieuHuy,LEN(MaPhieuHuy)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm phiếu tạo trong 7 ngày trước (không tính hôm nay)

\*/

public List<PhieuHuyHang> tim7NgayTruoc(String LoaiPhieuHuy) {

char x[] = LoaiPhieuHuy.toCharArray();

String LoaiPhieu = "";

for(int i = 0; i < x.length; i++){

if (i != x.length - 1)

LoaiPhieu += x[i] + ",";

else

LoaiPhieu += x[i];

}

String sql = "SELECT \* FROM PhieuHuyHang "

+ "WHERE LoaiPhieuHuy IN (" + LoaiPhieu + ") "

+ "AND NgayHuy < FORMAT(getdate(),'yyyy-MM-dd') "

+ "AND NgayHuy >= FORMAT(getdate()-7,'yyyy-MM-dd')"

+ "ORDER BY CAST(RIGHT(MaPhieuHuy,LEN(MaPhieuHuy)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm phiếu tạo hôm qua

\*/

public List<PhieuHuyHang> timNgayHomQua(String LoaiPhieuHuy) {

char x[] = LoaiPhieuHuy.toCharArray();

String LoaiPhieu = "";

for(int i = 0; i < x.length; i++){

if (i != x.length - 1)

LoaiPhieu += x[i] + ",";

else

LoaiPhieu += x[i];

}

String sql = "SELECT \* FROM PhieuHuyHang "

+ "WHERE LoaiPhieuHuy IN (" + LoaiPhieu + ") "

+ "AND NgayHuy = FORMAT(getdate()-1,'yyyy-MM-dd') "

+ "ORDER BY CAST(RIGHT(MaPhieuHuy,LEN(MaPhieuHuy)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm phiếu tạo trong ngày

\*/

public List<PhieuHuyHang> timTrongNgay(String LoaiPhieuHuy) {

char x[] = LoaiPhieuHuy.toCharArray();

String LoaiPhieu = "";

for(int i = 0; i < x.length; i++){

if (i != x.length - 1)

LoaiPhieu += x[i] + ",";

else

LoaiPhieu += x[i];

}

String sql = "SELECT \* FROM PhieuHuyHang "

+ "WHERE LoaiPhieuHuy IN (" + LoaiPhieu + ") "

+ "AND NgayHuy = FORMAT(getdate(),'yyyy-MM-dd') "

+ "ORDER BY CAST(RIGHT(MaPhieuHuy,LEN(MaPhieuHuy)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm theo mã sản phẩm hoặc tên sản phẩm

\*/

public List<PhieuHuyHang> timTheoSanPham(String ChuoiTimKiem) {

String sql = "SELECT \* FROM PhieuHuyHang PH "

+ "WHERE EXISTS "

+ "(SELECT MaPhieuHuy FROM PhieuHuyChiTiet PHCT "

+ "WHERE PHCT.MaPhieuHuy = PH.MaPhieuHuy "

+ "AND EXISTS "

+ "(SELECT MaSanPham FROM SanPham SP "

+ "WHERE PHCT.MaSanPham = SP.MaSanPham "

+ "AND (MaSanPham = '" + ChuoiTimKiem +"' OR TenSanPham LIKE N'%" + ChuoiTimKiem + "%'))) "

+ "ORDER BY CAST(RIGHT(MaPhieuHuy,LEN(MaPhieuHuy)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm theo mã chứng từ

\*/

public PhieuHuyHang timTheoMa(String MaChungTu){

String sql = "SELECT \* FROM PhieuHuyHang WHERE MaPhieuHuy = ?";

List<PhieuHuyHang> list = select(sql, MaChungTu);

return list.size() > 0 ? list.get(0) : null;

}

/\*

\* Hàm đếm tổng số phiếu huỷ

\*/

public int demTongPhieu() throws SQLException {

int dem = 0;

String sql = "SELECT COUNT(\*) AS 'COUNT' FROM PhieuHuyHang";

ResultSet rs = jdbcHelper.executeQuery(sql);

while(rs.next())

dem = rs.getInt("COUNT");

return dem;

}

/\*

\* Hàm tìm tất cả phiếu huỷ

\*/

public List<PhieuHuyHang> select(){

String sql = "SELECT \* FROM PhieuHuyHang "

+ "ORDER BY CAST(RIGHT(MaPhieuHuy,LEN(MaPhieuHuy)-2) AS int)";

return select(sql);

}

private List<PhieuHuyHang> select(String sql, Object... args) {

List<PhieuHuyHang> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

PhieuHuyHang model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private PhieuHuyHang readFromResultSet(ResultSet rs) throws SQLException {

PhieuHuyHang model = new PhieuHuyHang();

model.setMaPhieuHuy(rs.getString("MaPhieuHuy"));

model.setLoaiPhieuHuy(rs.getInt("LoaiPhieuHuy"));

model.setMaNhanVien(rs.getString("MaNhanVien"));

model.setNgayHuy(rs.getDate("NgayHuy"));

model.setSoLuongHuy(rs.getInt("SoLuongHuy"));

model.setGiaTriHuy(rs.getFloat("GiaTriHuy"));

model.setGhiChu(rs.getString("GhiChu"));

return model;

}}

* + - 1. **PhieuHuyChiTietDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import model.PhieuHuyChiTiet;

/\*\*

\*

\* @author Administrator

\*/

public class PhieuHuyChiTietDAO {

/\*

\* Insert phiếu huỷ chi tiết

\*/

public void themPHCT(PhieuHuyChiTiet model){

String sql = "INSERT INTO PhieuHuyChiTiet(MaPhieuHuy, MaSanPham, "

+ "DVT, SoLuong, GiaVon, ThanhTien) "

+ "VALUES (?, ?, ?, ?, ?, ?)";

jdbcHelper.executeUpdate(sql,

model.getMaPhieuHuy(),

model.getMaSanPham(),

model.getDVT(),

model.getSoLuong(),

model.getGiaVon(),

model.getThanhTien());

}

/\*

\* Tìm danh sách phiếu huỷ chi tiết theo mã phiếu huỷ

\*/

public List<PhieuHuyChiTiet> timDanhSachPHCT(String MaPhieuHuy){

String sql = "SELECT \* FROM PhieuHuyChiTiet WHERE MaPhieuHuy = ?";

return select(sql, MaPhieuHuy);

}

private List<PhieuHuyChiTiet> select(String sql, Object... args) {

List<PhieuHuyChiTiet> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

PhieuHuyChiTiet model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private PhieuHuyChiTiet readFromResultSet(ResultSet rs) throws SQLException {

PhieuHuyChiTiet model = new PhieuHuyChiTiet();

model.setMaPHCT(rs.getInt("MaPHCT"));

model.setMaPhieuHuy(rs.getString("MaPhieuHuy"));

model.setMaSanPham(rs.getString("MaSanPham"));

model.setDVT(rs.getString("DVT"));

model.setSoLuong(rs.getInt("SoLuong"));

model.setGiaVon(rs.getFloat("GiaVon"));

model.setThanhTien(rs.getFloat("ThanhTien"));

return model;

}}

* + - 1. **PhieuKiemDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import model.PhieuKiem;

/\*\*

\*

\* @author Administrator

\*/

public class PhieuKiemDAO {

/\*

\* Hàm thêm phiếu đã cân bằng

\*/

public int themPhieuCanBang(PhieuKiem model) {

String sql = "INSERT INTO PhieuKiemKe(MaPhieuKiemKe, MaNhanVien, NgayTao, NgayCanBang, "

+ "GiaTriLech, GhiChu, TrangThai) "

+ "VALUES (?, ?, ?, ?, ?, ?, ?)";

return jdbcHelper.executeUpdate(sql,

model.getMaPhieuKiem(),

model.getMaNhanVien(),

model.getNgayTao(),

model.getNgayCanBang(),

model.getGiaTriLech(),

model.getGhiChu(),

1);

}

/\*

\* Hàm thêm phiếu tạm

\*/

public int themPhieuTam(PhieuKiem model) {

String sql = "INSERT INTO PhieuKiemKe(MaPhieuKiemKe, MaNhanVien, NgayTao, "

+ "GiaTriLech, GhiChu, TrangThai) "

+ "VALUES (?, ?, ?, ?, ?, ?)";

return jdbcHelper.executeUpdate(sql,

model.getMaPhieuKiem(),

model.getMaNhanVien(),

model.getNgayTao(),

model.getGiaTriLech(),

model.getGhiChu(),

0);

}

/\*

\* Hàm cập nhật phiếu kiểm

\*/

public int capnhatPhieuKiem(PhieuKiem model) {

String sql = "{call sp\_update\_phieukiemke(?, ?, ?, ?)}";

return jdbcHelper.executeUpdate(sql,

model.getGiaTriLech(),

model.getGhiChu(),

model.getTrangThai(),

model.getMaPhieuKiem());

}

/\*

\* Hàm huỷ phiếu kiểm

\*/

public int huyPhieuKiem(PhieuKiem model) {

String sql = "UPDATE PhieuKiemKe SET TrangThai = 2, GhiChu = ? "

+ "WHERE MaPhieuKiemKe = ?";

return jdbcHelper.executeUpdate(sql,

model.getGhiChu(),

model.getMaPhieuKiem());

}

/\*

\* Hàm đếm tổng số phiếu kiểm

\*/

public int demPhieuKiem() throws SQLException {

int dem = 0;

String sql = "SELECT COUNT(\*) AS 'COUNT' FROM PhieuKiemKe";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next())

dem = rs.getInt("COUNT");

return dem;

}

/\*

\* Hàm tìm phiếu kiểm theo trạng thái

\*/

public List<PhieuKiem> timTheoTrangThai(String ChuoiTrangThai) {

char x[] = ChuoiTrangThai.toCharArray();

String TrangThai = "";

for(int i = 0; i < x.length; i++){

if (i != x.length - 1)

TrangThai += x[i] + ",";

else

TrangThai += x[i];

}

String sql = "SELECT \* FROM PhieuKiemKe "

+ "WHERE TrangThai IN (" + TrangThai + ") "

+ "ORDER BY CAST(RIGHT(MaPhieuKiemKe,LEN(MaPhieuKiemKe)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm phiếu kiểm theo ngày tuỳ chọn

\*/

public List<PhieuKiem> timTheoKhoangNgay(String ChuoiTrangThai, String TuNgay, String DenNgay) {

char x[] = ChuoiTrangThai.toCharArray();

String TrangThai = "";

for(int i = 0; i < x.length; i++){

if (i != x.length - 1)

TrangThai += x[i] + ",";

else

TrangThai += x[i];

}

String sql = "SELECT \* FROM PhieuKiemKe "

+ "WHERE TrangThai IN (" + TrangThai + ") "

+ "AND NgayTao >= '" + TuNgay + "' "

+ "AND NgayTao <= '" + DenNgay + "' "

+ "ORDER BY CAST(RIGHT(MaPhieuKiemKe,LEN(MaPhieuKiemKe)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm phiếu kiểm trong 7 ngày trước (không tính hôm nay)

\*/

public List<PhieuKiem> tim7NgayTruoc(String ChuoiTrangThai) {

char x[] = ChuoiTrangThai.toCharArray();

String TrangThai = "";

for(int i = 0; i < x.length; i++){

if (i != x.length - 1)

TrangThai += x[i] + ",";

else

TrangThai += x[i];

}

String sql = "SELECT \* FROM PhieuKiemKe "

+ "WHERE TrangThai IN (" + TrangThai + ") "

+ "AND NgayTao < FORMAT(getdate(),'yyyy-MM-dd') "

+ "AND NgayTao >= FORMAT(getdate()-7,'yyyy-MM-dd')"

+ "ORDER BY CAST(RIGHT(MaPhieuKiemKe,LEN(MaPhieuKiemKe)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm phiếu kiểm tạo hôm qua

\*/

public List<PhieuKiem> timNgayHomQua(String ChuoiTrangThai) {

char x[] = ChuoiTrangThai.toCharArray();

String TrangThai = "";

for(int i = 0; i < x.length; i++){

if (i != x.length - 1)

TrangThai += x[i] + ",";

else

TrangThai += x[i];

}

String sql = "SELECT \* FROM PhieuKiemKe "

+ "WHERE TrangThai IN (" + TrangThai + ") "

+ "AND NgayTao = FORMAT(getdate()-1,'yyyy-MM-dd') "

+ "ORDER BY CAST(RIGHT(MaPhieuKiemKe,LEN(MaPhieuKiemKe)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm phiếu kiểm tạo trong ngày

\*/

public List<PhieuKiem> timTrongNgay(String ChuoiTrangThai) {

char x[] = ChuoiTrangThai.toCharArray();

String TrangThai = "";

for(int i = 0; i < x.length; i++){

if (i != x.length - 1)

TrangThai += x[i] + ",";

else

TrangThai += x[i];

}

String sql = "SELECT \* FROM PhieuKiemKe "

+ "WHERE TrangThai IN (" + TrangThai + ") "

+ "AND NgayTao = FORMAT(getdate(),'yyyy-MM-dd') "

+ "ORDER BY CAST(RIGHT(MaPhieuKiemKe,LEN(MaPhieuKiemKe)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm theo mã sản phẩm hoặc tên sản phẩm

\*/

public List<PhieuKiem> timTheoSanPham(String ChuoiTimKiem) {

String sql = "SELECT \* FROM PhieuKiemKe PK "

+ "WHERE EXISTS "

+ "(SELECT MaPhieuKiem FROM PhieuKiemChiTiet PKCT "

+ "WHERE PKCT.MaPhieuKiem = PK.MaPhieuKiemKe "

+ "AND EXISTS "

+ "(SELECT MaSanPham FROM SanPham SP "

+ "WHERE PKCT.MaSanPham = SP.MaSanPham "

+ "AND (MaSanPham = '" + ChuoiTimKiem +"' OR TenSanPham LIKE N'%" + ChuoiTimKiem + "%'))) "

+ "ORDER BY CAST(RIGHT(MaPhieuKiemKe,LEN(MaPhieuKiemKe)-2) AS int)";

return select(sql);

}

/\*

\* Hàm tìm theo mã chứng từ

\*/

public PhieuKiem timTheoMa(String MaChungTu){

String sql = "SELECT \* FROM PhieuKiemKe WHERE MaPhieuKiemKe = ?";

List<PhieuKiem> list = select(sql, MaChungTu);

return list.size() > 0 ? list.get(0) : null;

}

/\*

\* Hàm tìm tất cả phiếu kiểm

\*/

public List<PhieuKiem> select(){

String sql = "SELECT \* FROM PhieuKiemKe "

+ "ORDER BY CAST(RIGHT(MaPhieuKiemKe,LEN(MaPhieuKiemKe)-2) AS int)";

return select(sql);

}

private List<PhieuKiem> select(String sql, Object... args) {

List<PhieuKiem> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

PhieuKiem model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private PhieuKiem readFromResultSet(ResultSet rs) throws SQLException {

PhieuKiem model = new PhieuKiem();

model.setMaPhieuKiem(rs.getString("MaPhieuKiemKe"));

model.setMaNhanVien(rs.getString("MaNhanVien"));

model.setNgayTao(rs.getDate("NgayTao"));

model.setNgayCanBang(rs.getDate("NgayCanBang"));

model.setGiaTriLech(rs.getDouble("GiaTriLech"));

model.setGhiChu(rs.getString("GhiChu"));

model.setTrangThai(rs.getInt("TrangThai"));

return model;

}}

* + - 1. **PhieuKiemChiTietDAO**

package dao;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

import model.PhieuKiemChiTiet;

/\*\*

\*

\* @author Administrator

\*/

public class PhieuKiemChiTietDAO {

/\*

\* Insert phiếu kiểm chi tiết

\*/

public void themPKCT(PhieuKiemChiTiet model){

String sql = "INSERT INTO PhieuKiemChiTiet(MaPhieuKiem, MaSanPham, DonGia, "

+ "DVT, TonKho, ThucTe, SoLuongLech, GiaTriLech) "

+ "VALUES (?, ?, ?, ?, ?, ?, ?, ?)";

jdbcHelper.executeUpdate(sql,

model.getMaPhieuKiem(),

model.getMaSanPham(),

model.getDonGia(),

model.getDVT(),

model.getTonKho(),

model.getThucTe(),

model.getSoLuongLech(),

model.getGiaTriLech());

}

/\*

\* Cập nhật phiếu kiểm chi tiết

\*/

public void capnhatPKCT(PhieuKiemChiTiet model){

String sql = "UPDATE PhieuKiemChiTiet SET "

+ "DonGia = ?, TonKho = ?, ThucTe = ?, "

+ "SoLuongLech = ?, GiaTriLech = ? "

+ "WHERE MaPKCT = ?";

jdbcHelper.executeUpdate(sql,

model.getDonGia(),

model.getTonKho(),

model.getThucTe(),

model.getSoLuongLech(),

model.getGiaTriLech(),

model.getMaPKCT());

}

/\*

\* Tìm danh sách phiếu kiểm chi tiết theo mã phiếu kiểm

\*/

public List<PhieuKiemChiTiet> timDanhSachPKCT(String MaPhieuKiem){

String sql = "SELECT \* FROM PhieuKiemChiTiet WHERE MaPhieuKiem = ?";

return select(sql, MaPhieuKiem);

}

private List<PhieuKiemChiTiet> select(String sql, Object... args) {

List<PhieuKiemChiTiet> list = new ArrayList<>();

try {

ResultSet rs = null;

try {

rs = jdbcHelper.executeQuery(sql, args);

while (rs.next()) {

PhieuKiemChiTiet model = readFromResultSet(rs);

list.add(model);

}

} finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException ex) {

throw new RuntimeException(ex);

}

return list;

}

private PhieuKiemChiTiet readFromResultSet(ResultSet rs) throws SQLException {

PhieuKiemChiTiet model = new PhieuKiemChiTiet();

model.setMaPKCT(rs.getInt("MaPKCT"));

model.setMaPhieuKiem(rs.getString("MaPhieuKiem"));

model.setMaSanPham(rs.getString("MaSanPham"));

model.setDonGia(rs.getFloat("DonGia"));

model.setDVT(rs.getString("DVT"));

model.setTonKho(rs.getInt("TonKho"));

model.setThucTe(rs.getInt("ThucTe"));

model.setSoLuongLech(rs.getInt("SoLuongLech"));

model.setGiaTriLech(rs.getDouble("GiaTriLech"));

return model;

}}

* + - 1. **ThongKeDAO**

package dao;

import helper.DateHelper;

import helper.jdbcHelper;

import java.sql.ResultSet;

import java.sql.SQLException;

import java.util.ArrayList;

import java.util.List;

/\*\*

\*

\* @author Administrator

\*/

public class ThongKeDAO {

/\*

\* Hàm lập bảng doanh thu 7 ngày gần nhất

\*/

public List<Object[]> bangDoanhThu7Ngay() throws SQLException {

List<Object[]> list = new ArrayList<>();

String sql = "{call sp\_bc\_doanhthu7ngay}";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

Object[] model = {rs.getDouble(1), rs.getDouble(2), rs.getDouble(3), rs.getDouble(4), rs.getDouble(5), rs.getDouble(6), rs.getDouble(7)};

list.add(model);

}

return list;

}

/\*

\* Hàm thống kê số sản phẩm sắp hết HSD trong 7 ngày tới

\*/

public int demSanPhamHetHSD() throws SQLException {

int dem = 0;

String sql = "SELECT COUNT(\*) AS 'DEM' FROM SanPham "

+ "WHERE HSDTruoc <= (getdate()+7) "

+ "AND HSDTruoc >= getdate()";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

dem = rs.getInt("DEM");

}

return dem;

}

/\*

\* Hàm tính giá vốn hàng bán theo khoảng thời gian

\*/

public double tinhGiaVonHangBan(String TuNgay, String DenNgay) {

double GiaVon = 0;

try {

ResultSet rs = null;

try{

String sql = "{call sp\_giavonhangban(?, ?)}";

rs = jdbcHelper.executeQuery(sql, DateHelper.toDate(TuNgay, "yyyy-MM-dd"), DateHelper.toDate(DenNgay, "yyyy-MM-dd"));

while(rs.next()){

GiaVon = rs.getDouble(1);

}

}

finally{

rs.getStatement().getConnection().close();

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

return GiaVon;

}

/\*

\* Hàm tính tổng số lượng hàng nhập theo thời gian

\*/

public int tinhTongSLHangNhap(String TuNgay, String DenNgay) throws SQLException {

int Tong = 0;

String sql = "SELECT SUM(TongSoLuong) AS 'Tong' FROM PhieuNhapHang "

+ "WHERE NgayNhap >= '" + TuNgay + "' AND NgayNhap <= '" + DenNgay + "'";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

Tong = rs.getInt("Tong");

}

return Tong;

}

/\*

\* Hàm tính tổng số lượng hàng nhập toàn thời gian

\*/

public int tinhTongSLHangNhap() throws SQLException {

int Tong = 0;

String sql = "SELECT SUM(TongSoLuong) AS 'Tong' FROM PhieuNhapHang";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

Tong = rs.getInt("Tong");

}

return Tong;

}

/\*

\* Hàm tính tổng giá trị hàng nhập theo thời gian

\*/

public double tinhTongGiaTriHangNhap(String TuNgay, String DenNgay) throws SQLException {

double Tong = 0;

String sql = "SELECT SUM(ThanhTien) AS 'Tong' FROM PhieuNhapHang "

+ "WHERE NgayNhap >= '" + TuNgay + "' AND NgayNhap <= '" + DenNgay + "'";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

Tong = rs.getDouble("Tong");

}

return Tong;

}

/\*

\* Hàm tính tổng giá trị hàng nhập toàn thời gian

\*/

public double tinhTongGiaTriHangNhap() throws SQLException {

double Tong = 0;

String sql = "SELECT SUM(ThanhTien) AS 'Tong' FROM PhieuNhapHang";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

Tong = rs.getDouble("Tong");

}

return Tong;

}

/\*

\* Hàm thống kê tổng hợp hàng nhập theo thời gian tuỳ chọn

\*/

public List<Object[]> thongkeHangNhap(String TuNgay, String DenNgay) {

List<Object[]> list = new ArrayList<>();

try {

ResultSet rs = null;

try{

String sql = "{call sp\_tk\_tonghophangnhap\_theotg(?, ?)}";

rs = jdbcHelper.executeQuery(sql, DateHelper.toDate(TuNgay, "yyyy-MM-dd"), DateHelper.toDate(DenNgay, "yyyy-MM-dd"));

while(rs.next()){

Object[] model = {rs.getString(1), rs.getString(2), rs.getString(3), rs.getInt(4), rs.getFloat(5)};

list.add(model);

}

}

finally{

rs.getStatement().getConnection().close();

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

return list;

}

/\*

\* Hàm thống kê tổng hợp hàng nhập toàn thời gian

\*/

public List<Object[]> thongkeHangNhapToanTG() {

List<Object[]> list = new ArrayList<>();

try {

ResultSet rs = null;

try{

String sql = "{call sp\_tk\_tonghophangnhap()}";

rs = jdbcHelper.executeQuery(sql);

while(rs.next()){

Object[] model = {rs.getString(1), rs.getString(2), rs.getString(3), rs.getInt(4), rs.getFloat(5)};

list.add(model);

}

}

finally{

rs.getStatement().getConnection().close();

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

return list;

}

/\*

\* Hàm thống kê hàng bán chạy theo thời gian tuỳ chọn

\*/

public List<Object[]> locSPBanChay(String TuNgay, String DenNgay) {

List<Object[]> list = new ArrayList<>();

try {

ResultSet rs = null;

try{

String sql = "{call sp\_tk\_hangbanchay(?, ?)}";

rs = jdbcHelper.executeQuery(sql, DateHelper.toDate(TuNgay, "yyyy-MM-dd"), DateHelper.toDate(DenNgay, "yyyy-MM-dd"));

while(rs.next()){

Object[] model = {rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4)};

list.add(model);

}

}

finally{

rs.getStatement().getConnection().close();

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

return list;

}

/\*

\* Hàm thống kê hàng bán chạy toàn thời gian

\*/

public List<Object[]> locSPBanChayToanThoiGian() {

List<Object[]> list = new ArrayList<>();

try {

ResultSet rs = null;

try{

String sql = "{call sp\_tk\_hangbanchay\_toantg()}";

rs = jdbcHelper.executeQuery(sql);

while(rs.next()){

Object[] model = {rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4)};

list.add(model);

}

}

finally{

rs.getStatement().getConnection().close();

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

return list;

}

/\*

\* Hàm thống kê tổng giá trị kho hàng

\*/

public double tongGiaTriKho() throws SQLException {

double Tong = 0;

String sql = "SELECT SUM(TonKho\*GiaVon) AS 'Tong' FROM SanPham";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

Tong = rs.getInt("Tong");

}

return Tong;

}

/\*

\* Hàm thống kê tổng số lượng sản phẩm trong kho hàng

\*/

public int demTongSanPham() throws SQLException {

int dem = 0;

String sql = "SELECT SUM(TonKho) AS 'Tong' FROM SanPham";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

dem = rs.getInt("Tong");

}

return dem;

}

/\*

\* Hàm liệt kê thông tin kho hàng

\*/

public List<Object[]> thongkeKhoHang() {

List<Object[]> list = new ArrayList<>();

try {

ResultSet rs = null;

try{

String sql = "SELECT MaNhomHang, MaSanPham, TenSanPham, DVT.MoTa, "

+ "TonKho, GiaVon, (GiaVon\*TonKho) AS 'GiaTri' "

+ "FROM SanPham LEFT JOIN DonViTinh DVT "

+ "ON SanPham.MaDVT = DVT.MaDVT "

+ "WHERE TonKho > 0 "

+ "ORDER BY CAST(RIGHT(MaNhomHang, LEN(MaNhomHang)-2) AS int), "

+ "CAST(RIGHT(MaSanPham, LEN(MaSanPham)-2) AS int)";

rs = jdbcHelper.executeQuery(sql);

while(rs.next()){

Object[] model = {rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getInt(5), rs.getFloat(6), rs.getFloat(7)};

list.add(model);

}

}

finally{

rs.getStatement().getConnection().close();

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

return list;

}

/\*

\* Hàm tính số hoá đơn bán được trong năm

\*/

public int demSoDonTrongNam() throws SQLException {

int SoHD = 0;

String sql = "SELECT COUNT(\*) AS 'Tong' FROM HoaDon "

+ "WHERE YEAR(NgayBan) = YEAR(getdate())";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

SoHD = rs.getInt("Tong");

}

return SoHD;

}

/\*

\* Hàm tính doanh thu trong năm

\*/

public double tinhDoanhThuTrongNam() throws SQLException {

double DoanhThu = 0;

String sql = "SELECT SUM(ThanhTien) AS 'Tong' FROM HoaDon "

+ "WHERE YEAR(NgayBan) = YEAR(getdate())";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

DoanhThu = rs.getDouble("Tong");

}

return DoanhThu;

}

/\*

\* Hàm tính số hoá đơn bán được trong quý

\*/

public int demSoDonTrongQuy() throws SQLException {

int Quy = DateHelper.tinhSoQuyTrongNam();

int SoHD = 0;

String sql = "SELECT COUNT(\*) AS 'Tong' FROM HoaDon "

+ "WHERE YEAR(NgayBan) = YEAR(getdate()) ";

switch (Quy) {

case 1:

sql+= "AND MONTH(NgayBan) >= 1 AND MONTH(NgayBan) <= 3";

break;

case 2:

sql+= "AND MONTH(NgayBan) >= 4 AND MONTH(NgayBan) <= 6";

break;

case 3:

sql+= "AND MONTH(NgayBan) >= 7 AND MONTH(NgayBan) <= 9";

break;

case 4:

sql+= "AND MONTH(NgayBan) >= 10 AND MONTH(NgayBan) <= 12";

break;

}

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

SoHD = rs.getInt("Tong");

}

return SoHD;

}

/\*

\* Hàm tính doanh thu trong quý

\*/

public double tinhDoanhThuTrongQuy() throws SQLException {

int Quy = DateHelper.tinhSoQuyTrongNam();

double DoanhThu = 0;

String sql = "SELECT SUM(ThanhTien) AS 'Tong' FROM HoaDon "

+ "WHERE YEAR(NgayBan) = YEAR(getdate()) ";

switch (Quy) {

case 1:

sql+= "AND MONTH(NgayBan) >= 1 AND MONTH(NgayBan) <= 3";

break;

case 2:

sql+= "AND MONTH(NgayBan) >= 4 AND MONTH(NgayBan) <= 6";

break;

case 3:

sql+= "AND MONTH(NgayBan) >= 7 AND MONTH(NgayBan) <= 9";

break;

case 4:

sql+= "AND MONTH(NgayBan) >= 10 AND MONTH(NgayBan) <= 12";

break;

}

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

DoanhThu = rs.getDouble("Tong");

}

return DoanhThu;

}

/\*

\* Hàm tính doanh thu trong tháng

\*/

public double tinhDoanhThuTrongThang() throws SQLException {

double DoanhThu = 0;

String sql = "SELECT SUM(ThanhTien) AS 'Tong' FROM HoaDon "

+ "WHERE MONTH(NgayBan) = MONTH(getdate()) "

+ "AND YEAR(NgayBan) = YEAR(getdate())";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

DoanhThu = rs.getDouble("Tong");

}

return DoanhThu;

}

/\*

\* Hàm tính số hoá đơn bán được trong tháng

\*/

public int demSoDonTrongThang() throws SQLException {

int SoHD = 0;

String sql = "SELECT COUNT(\*) AS 'Tong' FROM HoaDon "

+ "WHERE MONTH(NgayBan) = MONTH(getdate()) "

+ "AND YEAR(NgayBan) = YEAR(getdate())";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

SoHD = rs.getInt("Tong");

}

return SoHD;

}

/\*

\* Hàm tính số hoá đơn bán được trong ngày

\*/

public int demSoDonTrongNgay() throws SQLException {

int SoHD = 0;

String sql = "SELECT COUNT(\*) AS 'Tong' FROM HoaDon "

+ "WHERE NgayBan = FORMAT(getdate(), 'yyyy-MM-dd')";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

SoHD = rs.getInt("Tong");

}

return SoHD;

}

/\*

\* Hàm tính số hoá đơn bán được trong tuần

\*/

public int demSoDonTrongTuan() throws SQLException {

String DauTuan = DateHelper.tinhNgayDauTuan(1);

int SoHD = 0;

String sql = "SELECT COUNT(\*) AS 'Tong' FROM HoaDon "

+ "WHERE NgayBan <= FORMAT(getdate(), 'yyyy-MM-dd') "

+ "AND NgayBan >= '" + DauTuan + "'";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

SoHD = rs.getInt("Tong");

}

return SoHD;

}

/\*

\* Hàm tính doanh thu trong tuần đến ngày hiện tại

\*/

public double tinhDoanhThuTrongTuan() throws SQLException {

String DauTuan = DateHelper.tinhNgayDauTuan(1);

double DoanhThu = 0;

String sql = "SELECT SUM(ThanhTien) AS 'Tong' FROM HoaDon "

+ "WHERE NgayBan <= FORMAT(getdate(), 'yyyy-MM-dd') "

+ "AND NgayBan >= '" + DauTuan + "'";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

DoanhThu = rs.getDouble("Tong");

}

return DoanhThu;

}

/\*

\* Hàm tính doanh thu trong ngày

\*/

public double tinhDoanhThuTrongNgay() throws SQLException {

double DoanhThu = 0;

String sql = "SELECT SUM(ThanhTien) AS 'Tong' FROM HoaDon "

+ "WHERE NgayBan = FORMAT(getdate(), 'yyyy-MM-dd')";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

DoanhThu = rs.getDouble("Tong");

}

return DoanhThu;

}

/\*

\* Hàm tính tổng doanh thu của cửa hàng theo toàn thời gian

\*/

public double tinhTongDoanhThu() throws SQLException {

double TongDoanhThu = 0;

String sql = "SELECT SUM(ThanhTien) AS 'Tong' FROM HoaDon";

ResultSet rs = jdbcHelper.executeQuery(sql);

while (rs.next()) {

TongDoanhThu = rs.getDouble("Tong");

}

return TongDoanhThu;

}

/\*

\* Hàm tính doanh thu của nhân viên theo mã nhân viên

\*/

public double tinhDoanhThuNV(String MaNV) throws SQLException {

double DoanhThu = 0;

String sql = "SELECT SUM(ThanhTien) AS 'DoanhThu' FROM HoaDon "

+ "WHERE MaNhanVien = ? "

+ "GROUP BY MaNhanVien";

ResultSet rs = jdbcHelper.executeQuery(sql, MaNV);

while(rs.next()) {

DoanhThu = rs.getDouble("DoanhThu");

}

return DoanhThu;

}

/\*

\* Hàm thống kê tổng tiền nhập và tổng tiền trả của nhà cung cấp

\*/

public List<Object[]> tongGiaoDichNCC(String MaNCC) throws SQLException{

List<Object[]> list = new ArrayList<>();

try {

ResultSet rs = null;

try{

String sql = "{call sp\_tonggiaodich\_ncc(?)}";

rs = jdbcHelper.executeQuery(sql, MaNCC);

while(rs.next()){

Object[] model = {rs.getString(1), rs.getFloat(2), rs.getFloat(3)};

list.add(model);

}

}

finally{

rs.getStatement().getConnection().close();

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

return list;

}

public float tongGiaoDichKH(String MaKhachHang) throws SQLException{

float TongGiaoDich = 0;

String sql = "SELECT SUM(ThanhTien) FROM HoaDon HD "

+ "JOIN KhachHang KH ON HD.MaKhachHang = KH.MaKhachHang "

+ "WHERE HD.MaKhachHang = ?";

ResultSet rs = null;

rs = jdbcHelper.executeQuery(sql, MaKhachHang);

while(rs.next()){

TongGiaoDich = rs.getFloat(1);

}

return TongGiaoDich;

}

/\*

\* Tổng kết cuối ngày: danh sách sản phẩm bán ra, số lượng và giá trị Thống kê theo nhân viên

\*/

public List<Object[]> ThongKeCuoiNgay(String MaNhanVien){

List<Object[]> list = new ArrayList<>();

String NgayBan = DateHelper.toString(DateHelper.now(), "yyyy-MM-dd");

try {

ResultSet rs = null;

try {

String sql = "{call sp\_tongketcuoingay(?, ?)}";

rs = jdbcHelper.executeQuery(sql, MaNhanVien, NgayBan);

while(rs.next()){

Object[] model = {rs.getString(1), rs.getInt(2), rs.getFloat(3)};

list.add(model);

}

}

finally {

rs.getStatement().getConnection().close();

}

} catch (SQLException e) {

throw new RuntimeException(e);

}

return list;

}}

**LỜI CẢM ƠN**

Trong quãng thời gian hơn một tháng qua, chúng em đã nhận được sự hỗ trợ tận tình của thầy **ThS. Lê Anh Tú –** giảng viên đứng lớp môn *“Dự án 1” –* để chúng em hoàn thiện phần mềm này.

Chúng em xin gởi lời cảm ơn đến thầy và chúc thầy nhiều sức khoẻ để tiếp tục hướng dẫn, hỗ trợ cho sinh viên Fpoly đạt được kết quả học tập tốt nhất.

Trân trọng

*Nhóm thực hiện đề tài*